Meghana Ravishankar

Project 2

Effect of Elbow Method on Clustering method using Devanagiri Dataset.

Table of contents

Introduction:

The dataset Devanagiri Handwritten Character Dataset can be found on UCI ML Repository(https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset)

in the raw image form, however the CSV format of this file was used which was found on kaggle(https://www.kaggle.com/rishianand/devanagari-character-set). The dataset consists of 92000 rows (sample images), and 1025 columns ie values of each pixel (pixel0000" to "pixel1023") in greyscale values (0 to 255). The column "character" represents the Devanagari character name corresponding to each image. Character is centered within 28 by 28 pixel, padding of 2 pixel is added on all four sides of actual character.

Note: The Dataset does not contain vowels, it consists of only consonants and numbers.

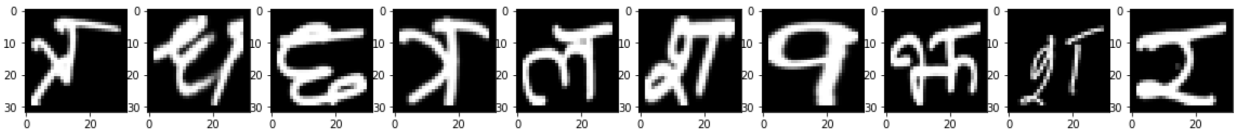

Figure 1: Sample of the Dataset

The aim of this project is study multiple pre clustering methods, after using the elbow method for pre clustering.

Methods:

1. Pre-Clustering

   The elbow method was used for pre clustering of the data. This method of interpretation and validation of consistency within clusters helps to find the appropriate number of clusters in a dataset. The idea of the elbow method is to run k-means clustering on the dataset for a range of values of k(number of Cluster to form) and for each value of k calculate the sum of squared errors (SSE). The algorithm is as follows:

   1. Compute k-means clustering for different values of k. For instance, by varying k from 1 to n clusters.
   2. For each k, calculate the total within-cluster sum of square (wss).
   3. Plot the curve of wss according to the number of clusters k.
   4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

2. Clustering

   K-means clustering is unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The algorithm finds groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically.

Mean shift is a feature-space analysis technique for locating the maxima of a density function. Application of this algorithm is in cluster analysis in computer vision and image processing. If you consider a set of points in two-dimensional space assuming a circular window centered at C and having radius r as the kernel. Mean shift is a hill climbing algorithm which involves shifting this kernel iteratively to a higher density region until convergence. The shift is defined by a mean shift vector which always points toward the direction of the maximum increase in the density. At every iteration the kernel is shifted to the centroid or the mean of the points within it. The method of calculating this mean depends on the choice of the kernel. At convergence, there will be no direction at which a shift can accommodate more points inside the kernel.

3. Post Clustering (Validation):

   Silhouette value measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates object is well matched to its own cluster,If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.The silhouette can be calculated with any distance metric,eg Euclidean distance or the Manhattan distance.Average silhouette method computes the average silhouette of observations for different values of k(K of K-means clustering ie number of clusters). The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k (Kaufman and Rousseeuw 1990).

Results:

<u>Pre-Clustering: Elbow method</u>

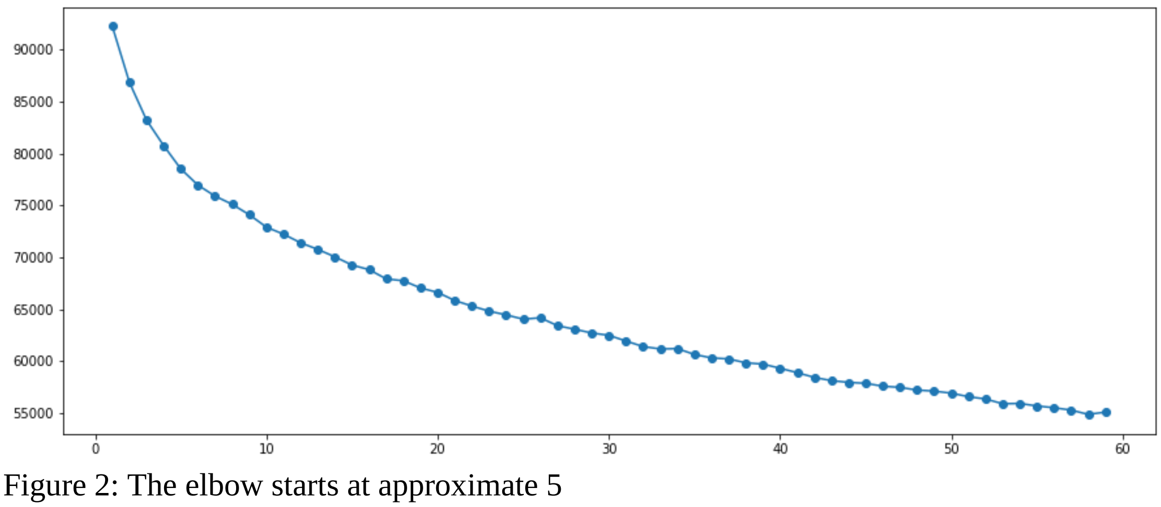

Figure 2: The elbow starts at approximate 5

The figure is a plot of number of ‘K’(ie clusters) vs within-cluster sum of square (WSS).From the figure we see that the curve bends at 5, thus we know that that the optimal cluster number to use in KMeans. The result obtained from using this method is used to pre define number of clusters for clustering algorithms. The dataset was subjected to Kmean, Meanshift and Hierarchical clustering.

Clustering :PCA-K mean

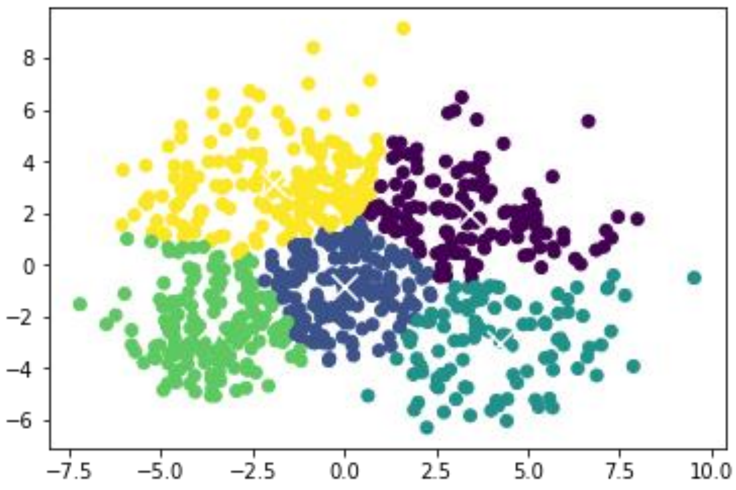

Figure 3

The average silhouette_score is : 0.371588134072

The cluster number was chosen as 5 from the results of the elbow method. Kmean was done on a PCA reduced data(PCA was done as a feature extraction, so that displaying number of clusters visulally is easier). PCA is NOT a effective feature extraction method for this dataset, as this is an image converted to CSV file data, Filtering would have been a better Feature extraction. Filtering could be done to find edges and curves.

Clustering :PCA-Agglomerative Clustering

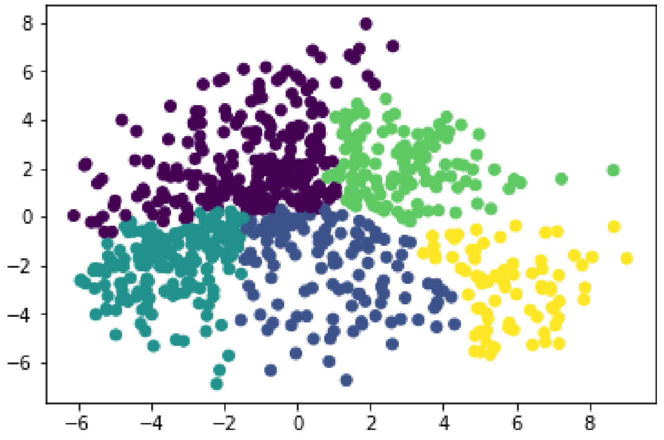

Figure 5

The average silhouette_score for AC is : 0.0368168895633

Clustering :PCA-MeanShift

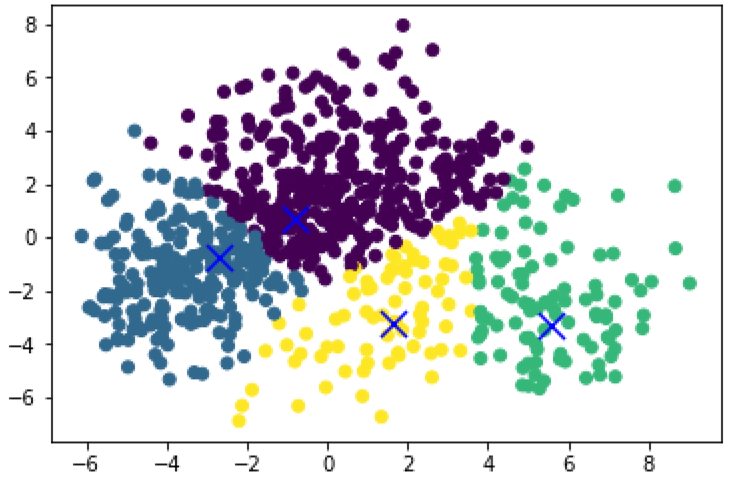

Figure 6

The average silhouette_score for MS is : 0.374740492942

Unlike in first 2 clustering methods, Meanshift does not need clusters to be defined.It clusters using Bandwidth estimation, which dictates the size of the region to search through. This parameter can be set manually.

Discussions:

From the results of cluster validation we can see that Mean Shift worked the best, and it did not also need Pre clustering, thus it saves a lot of time.

Why Mean Shift worked the best? Mean Shift does not make assumptions about the number of clusters or the shape of the cluster, therefore its ideal for handling clusters of arbitrary shape and number.Mean Shift is primarily a mode finding algorithm the stationary points obtained via gradient ascent represent the modes of the density function and all points associated with the same stationary point belong to the same cluster.

K-means is very sensitive to initializations. A wrong initialization can delay convergence or some times even result in wrong clusters. Mean shift is fairly robust to initializations.

Conclusion:

The Project introduced clustering algorithms and introduced pre clustering method which is the most important step for any unsupervised method. There are more algorithms for pre clustering to explore like GAP and multiple index methods. Feature extraction methods for image datasets include lot of filtering, which is basically convolutions, thus CNN works the best when working with image dataset and gives better accuracy. The project can be further continued to feed the data into a neural network and compare with clustering algorithms.

References:

1. http://www.recurrentconvolutions.com/going-beyond-mnist/

2. http://deciphertoknow.com/mnist-k-means-clustering/
3. http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods/
4. https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b
5. https://www.datacamp.com/community/tutorials/machine-learning-python

Appendix:

Code:

```
#preprocessing
import random
import numpy as np
import pandas as pd

#visualization
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

#load data
df = pd.read_csv(r"C:\Users\meghs\Documents\Big data\Project 2\DevanagariHandwrittenCharacterDataset\data.csv")
df=df.iloc[:10001,:]

#Seperate data from labels
data=df.iloc[:,0:1024]

#Labels
data_labels=df.iloc[:,-1]
data_label=np.unique(data_labels)

#taking only sample of data
data_sample=data.sample(750) #use random_state

#Converting the Pandas frame to numpy
data_np = np.array(data_sample.iloc[:,:],np.float32)
#Visualize how image looks
def visualize_input(img, ax):
    ax.imshow(img, cmap='gray')
    width, height = img.shape
    thresh = img.max()/2.5
    for x in range(width):
        for y in range(height):
            ax.annotate(str(round(img[x][y],2)), xy=(y,x),
                        horizontalalignment='center',
                        verticalalignment='center',
                        color='white' if img[x][y]<thresh else 'black')

fig = plt.figure(figsize = (12,12))
ax = fig.add_subplot(111)
visualize_input(data_np[156].reshape(32,32), ax)
from sklearn.cluster import KMeans
wcss = list() #within-cluster sum of square (WSS)
for k in range(1,60):
    kmeans = KMeans(n_clusters=k)
    kmeans = kmeans.fit(data_sample)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(15, 6))
plt.plot(range(1, 60), wcss, marker = "o")
n_clusters=5

reduced_data = PCA(n_components=2).fit_transform(data_sample)
kmeans = KMeans(init='k-means++', n_clusters=n_clusters, n_init=10)
k_pca=kmeans.fit(reduced_data)
Z = kmeans.predict(reduced_data)


plt.scatter(reduced_data[:, 0],reduced_data[:, 1],c=Z)
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1],
            marker='x', s=169, linewidths=3,
            color='b', zorder=10)
plt.show()

silhouette_avg = silhouette_score(reduced_data,Z)
print("For n_clusters =", n_clusters,
        "The average silhouette_score is :", silhouette_avg)
import numpy as np
from sklearn.cluster import MeanShift, estimate_bandwidth
```

```
X= reduced_data
bandwidth = estimate_bandwidth(X, quantile=0.09)

ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
ms.fit(X)
labels = ms.labels_
cluster_centers = ms.cluster_centers_
Z_ = ms.fit_predict(reduced_data)



labels_unique = np.unique(labels)
n_clusters_ = len(labels_unique)

print("number of estimated clusters : %d" % n_clusters_)

plt.scatter(reduced_data[:, 0],reduced_data[:, 1],c=Z_)
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
            marker='x', s=169, linewidths=3,
            color='b', zorder=10)
plt.show()



silhouette_avg_ms = silhouette_score(reduced_data,Z_)
print("For n_clusters =", n_clusters,
      "The average silhouette_score for MS is :", silhouette_avg_ms)
```