Implementing K-Means Clustering

Rakib Hossain Rifat
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
160204099@aust.edu

***Abstract*—Main objective of this experiment is implementing K-Means Clustering algorithm.**

***Index Terms*—KNN.**

I. Introduction

K-Means Clustering is used to classify data points and create n clusters based on the distance of the centroids using the given dataset.

II. Methodology

In this experiment I need to perform several tasks.

A. Plotting cluster based on given data

For this I need to plot training data of both cluster from ”data k mean.txt”, and samples belonging to same cluster should have same marker and color.

B. Perform K-Means Clustering

The value of K will be taken from user. Classify the clusters “test.txt” with different colored markers according to the predicted class label.while classifying Euclidean distance will be taken as a distance measure

for classifying data set given equation will be used ,

$$\sqrt{(X1-T1)^2+(X2-T2)^2}$$

C. Print new cluster with labels

After implementing K-Means Clustering algorithm new clusters with changed data points will be plotted.

III. Result Analysis

A. Plotting cluster based on given data

Here I have plotted data of both cluster with different marker and color.

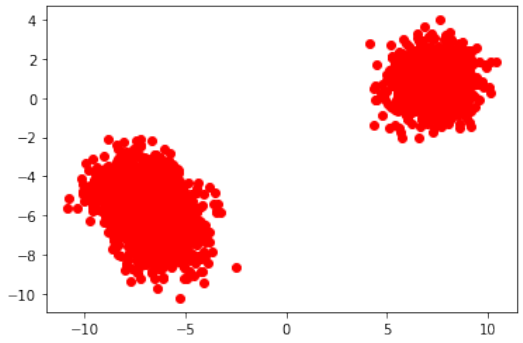

Fig. 1. cluster based on given data

B. New cluster with labels

Here I have plotted data of new cluster with different marker and color after implementing K-Means Clustering algorithm .

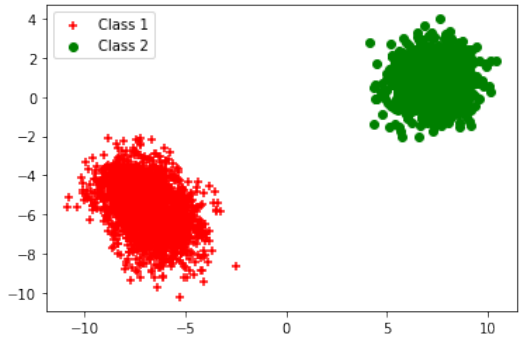

Fig. 2. New cluster with labels

IV. Conclusion

From the experiment it can be stated that data points can be classified using K-Means Clustering algorithm correctly.

V. Code

```
# -*- coding: utf-8 -*-
"""160204099_B2_05.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/13
    B5x6iKIcqHWjoPDT_iqp-P6jgP1bWEI
"""

import io
import pandas as pd


data = pd.read_csv('/content/data_k_mean.txt', sep =
     ' ', header = None)
data.head()
data_np = data.to_numpy()

import matplotlib.pyplot as plt
plt.scatter(data[0], data[1], c = 'r', marker = 'o')
plt.show()

k = int(input("Enter the value of k : "))

import numpy as np
np.random.seed(seed=4)
random_numbers = np.random.randint(low=0, high=len(
    data_np), size=(k,))
centroids = [data_np[random_numbers[i]] for i in
    range(k)]

distance = []
index_clusters = [-1 for i in range(len(data_np))]
count = 0
clusters = {}
flag=1
for x in range(200):
  count = x
  flag = 0
  for y in range(k):
    clusters[y] = []
  for i in range(len(data_np)):
    distance = []
    for j in range(k):
      dist = np.sqrt(pow((data_np[i][0] - centroids[
    j][0]), 2) + pow((data_np[i][1] - centroids[j
    ][1]), 2))
      distance.append(dist)
    ind = distance.index(min(distance))
    if index_clusters[i] != ind:
      flag = 1
      index_clusters[i] = ind
    clusters[ind].append(data_np[i])
  if flag == 0:
    break

  centroids = [np.mean(np.asarray(clusters[z]), axis
    =0) for z in range(k)]

x1 = np.asarray(clusters[0])[:, 0]
y1 = np.asarray(clusters[0])[:, 1]
x2 = np.asarray(clusters[1])[:, 0]
y2 = np.asarray(clusters[1])[:, 1]

plt.scatter(x1, y1, c = 'r', marker = '+', label = '
    Class 1')
plt.scatter(x2, y2, c = 'g', marker = 'o', label = '
    Class 2')
plt.legend(loc = 'best')
plt.show()
```