K Centre Algorithm

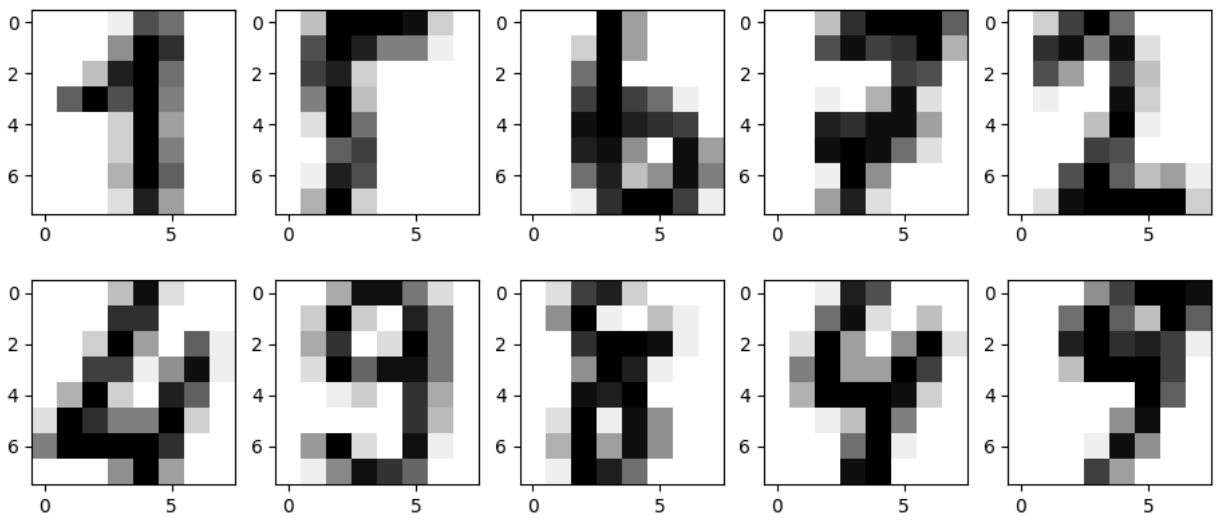
```
from matplotlib.pyplot import *
from sklearn.datasets import load_digits
import numpy as np
dig = load_digits()
def kcenter(data, k):
    cost = []
    c = []
    n = data.shape[0]
    # randomly initialize first center
    newc = data[np.random.randint(0,n)]
    c.append(newc)
    data2d = data[:,np.newaxis,:]
    for i in range(k):
        # calculating distances to new center
        newd = np.linalg.norm(data2d - newc[np.newaxis,:], axis=-1)
        if(i==0):
            distances = newd
        else :
            distances = np.concatenate((distances,newd),axis=1)
        # assign each point to the closest centre
        labels = np.argmin(distances, axis=1)
        # finding new center
        r = 0
        ci = "Not defined"
        for j in range(i+1):
            clusdists = distances[labels==j]
            rnew = np.amax(clusdists[:,j])
            if rnew > r :
                r = rnew
                ci = j
        clus = (labels==ci)
        clusdists = distances[clus][:,ci]
        clusdata = data[clus]
        newcdi = np.argmax(clusdists)
        newc = clusdata[newcdi]
        cost.append(r)
        c.append(newc)
    return c, labels, cost

#just for comparision
def kmean(data, k, updates=100):
    cost = []
    n = data.shape[0]
    # randomly initialize k centroids
    centroids = data[np.random.choice(n, k, replace=False)]
    for _ in range(updates):
        # assign each point to the closest centroid
```

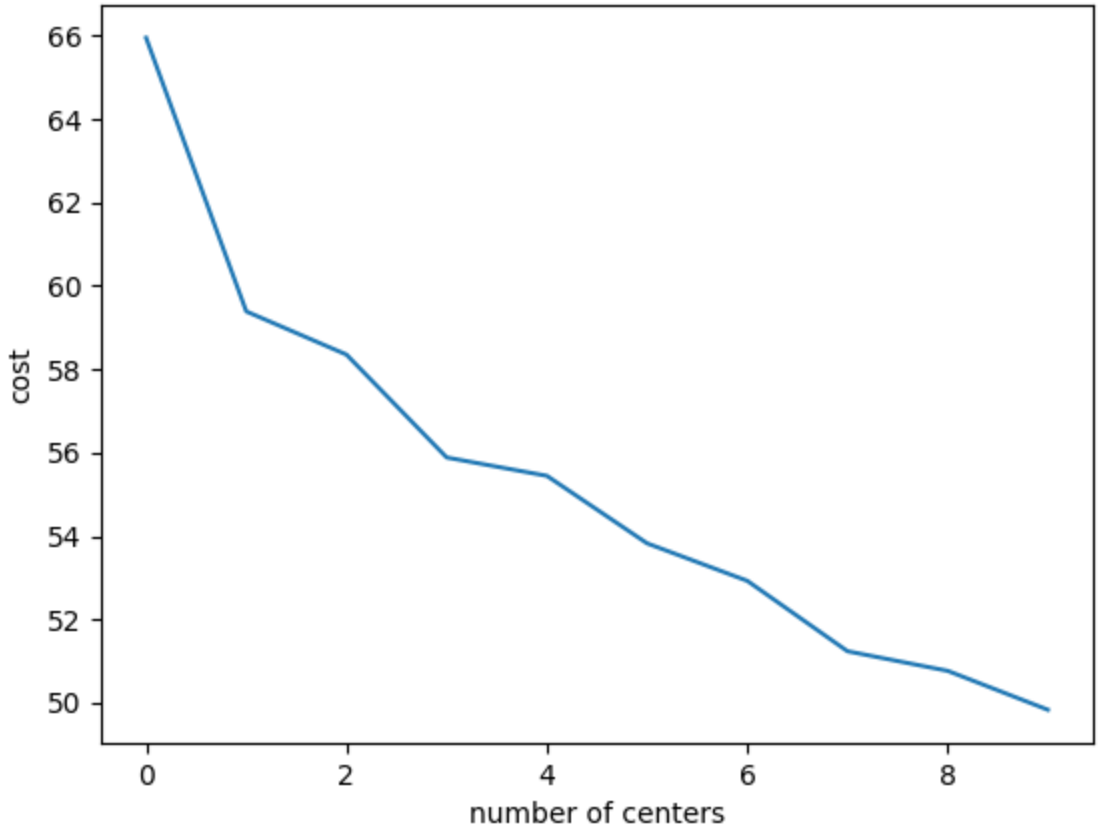
```
        distances = np.linalg.norm(data[:, np.newaxis, :] - centroids, axis=-1)
        labels = np.argmin(distances, axis=1)
        #calculating cost
        c = (np.min(distances,axis=1))**2
        cost.append(np.sum(c))
        # update centroids
        for i in range(k):
            centroids[i] = np.mean(data[labels == i], axis=0)
    return centroids, labels, cost

C,labels,cost = kcenter(dig["data"],10)
C2,labels2,cost2 = kmean(dig["data"],10)
M = np.zeros((10,10))
for i in range(labels.shape[0]):
    M[labels[i],labels2[i]] += 1
m,n = M.shape
for i in range(m):
    for j in range(n):
        text(j+0.5,m-i-0.5,str(M[i][j]),va="center",ha="center")
xlim([0,m])
ylim([0,n])
xticks(np.arange(n)+0.5,labels=list(map(str,range(n))))
yticks(np.arange(m)+0.5,labels=list(map(str,range(m))))
figure()
for i in range(10):
    subplot(2,5,i+1)
    im = np.split(C[i],8)
    imshow(im,"gray_r")
figure()
plot(np.arange(10),cost)
show()
print(cost[-1])
```

Final Output



Variation of cost with number of centres



Final cost : 49.82971001320397

Comparison with K means clusters