HW6_Problem1_Kmeans

May 21, 2021

[14]:
```
import numpy as np

# dataset
S = np.array([[0,0],[2,2],[0,1],[1,1]])
print('dataset:',S)
# initial mean
mu1, mu2 = [-1,-1], [3,3]
print('mu1:',mu1,';mu2:',mu2)
# 2 classes
c1, c2 = [],[]

# EM algorithm for K-means
iteration = 1
for i in range(iteration):
    print('---------',i+1,' iteration----------')
    # get distance between data and mean
    D1 = np.round(np.sqrt(np.sum((S-mu1)**2,axis=1)),2)
    D2 = np.round(np.sqrt(np.sum((S-mu2)**2,axis=1)),2)
    print('distance with mu1:',D1,';distance with mu2:',D2)
    [c1.append(S[j]) if D1[j] <= D2[j] else c2.append(S[j]) for j in␣
 ↪range(len(D1)) ]
    print('c1:',c1,';c2:',c2)
    # calculate new means
    new_mu1 = np.sum(c1, axis=0)/len(c1)
    new_mu2 = np.sum(c2, axis=0)/len(c2)
    print('new_mu1:',np.round(new_mu1,2),';new_mu2:',np.round(new_mu2,2))
```

```
dataset: [[0 0]
 [2 2]
 [0 1]
 [1 1]]
mu1: [-1, -1] ;mu2: [3, 3]
--------- 1  iteration----------
distance with mu1: [1.41 4.24 2.24 2.83] ;distance with mu2: [4.24 1.41 3.61
2.83]
c1: [array([0, 0]), array([0, 1]), array([1, 1])] ;c2: [array([2, 2])]
new_mu1: [0.33 0.67] ;new_mu2: [2. 2.]
```