

# DBSCAN

```
1 import numpy as np
2
3 Train_data = np.array([
4     [0.697, 0.460], [0.774, 0.376], [0.634, 0.264], [0.608, 0.318],
5     [0.556, 0.215],
6     [0.403, 0.237], [0.481, 0.149], [0.437, 0.211], [0.666, 0.091],
7     [0.243, 0.267],
8     [0.245, 0.057], [0.343, 0.099], [0.639, 0.161], [0.657, 0.198],
9     [0.360, 0.370],
10    [0.593, 0.042], [0.719, 0.103], [0.359, 0.188], [0.339, 0.241],
11    [0.282, 0.257],
12    [0.748, 0.232], [0.714, 0.346], [0.483, 0.312], [0.478, 0.437],
13    [0.525, 0.369],
14    [0.751, 0.489], [0.532, 0.472], [0.473, 0.376], [0.725, 0.445],
15    [0.446, 0.459]])
16 Train_size = Train_data.shape[0]
17
18
19
20 epsilon = 0.11
21 MinPts = 5
22 used = [False for i in range(Train_size)]
23 Kernel = []
24 isKernel = [False for i in range(Train_size)]
25
26 def distCmp(x, y):
27     return np.linalg.norm(x - y) <= epsilon
28
29 def Func1(*args):
30     global cnt
31     cnt += 1
32
33 def Func2(x):
34     # print(args)
35     if not used[x]:
36         if isKernel[x]:
37             Queue.append(x)
38             Kernel.remove(x)
39             C[-1].append(Train_data[x])
40             used[x] = True
41
42 def Judge(i, Func):
43     for j in range(Train_size):
44         if distCmp(Train_data[i], Train_data[j]):
45             Func(j)
46
47 for i in range(Train_size):
48     cnt = 0
49     Judge(i, Func1)
```

```
44     if cnt >= MinPts:
45         Kernel.append(i)
46         isKernel[i] = True
47 C = []
48 while Kernel:
49     Start = np.random.choice(Kernel, 1)[0]
50     used[Start] = True
51     Kernel.remove(Start)
52     Queue = [Start]
53     C.append([])
54     C[-1].append(Train_data[Start])
55     while Queue:
56         now = Queue[0]
57         Queue.pop(0)
58         Judge(now, Func2)
59
60 for x in C:
61     for y in x:
62         print(y)
63     print('\n')
64
65 for idx in range(Train_size):
66     if not used[idx]:
67         print('%d point is error' % idx)
```

运行结果

```

[ 0.483  0.312] cno
[ 0.478  0.437] String
[ 0.525  0.369] 麒麟
[ 0.446  0.459] 信息安全作品赛
[ 0.532  0.472] 信息安全作品赛
[ 0.608  0.318] OneDrive
[ 0.437  0.211] WPS网盘
[ 0.403  0.237] 最近
[ 0.481  0.149] 星标
[ 0.359  0.188] 我的云文档
[ 0.339  0.241] 共享文件夹
[ 0.343  0.099] 共享文件夹
[ 0.282  0.257] 此电脑
[ 0.243  0.267] 此电脑
[ 0.666  0.091] 3D 对象
[ 0.639  0.161] 视频
[ 0.657  0.198] 图片
[ 0.593  0.042] 文档
[ 0.719  0.103] 下载
[ 0.634  0.264] 音乐
[ 0.556  0.215] 桌面
[ 0.748  0.232] Windows (C:)
[ 0.725  0.445] DATA (D:)
[ 0.697  0.46 ] DATA (E:)
[ 0.774  0.376] 软件 (F:)
[ 0.714  0.346] 网络
[ 0.751  0.489] 网络

10 point is error
14 point is error

```

## kmeans

```

1  import numpy as np
2
3  k = 3
4
5  def rand_row(dataset, size):
6      n_size = dataset.shape[0]
7      row = np.random.choice(n_size, size, replace=False)
8      TrainMatrix = dataset[row,:]
9      return TrainMatrix
10
11
12
13  Train_data = np.array([
14      [0.697, 0.460], [0.774, 0.376], [0.634, 0.264], [0.608, 0.318],
15      [0.556, 0.215],
16      [0.403, 0.237], [0.481, 0.149], [0.437, 0.211], [0.666, 0.091],
17      [0.243, 0.267],

```

```

16     [0.245, 0.057], [0.343, 0.099], [0.639, 0.161], [0.657, 0.198],
    [0.360, 0.370],
17     [0.593, 0.042], [0.719, 0.103], [0.359, 0.188], [0.339, 0.241],
    [0.282, 0.257],
18     [0.748, 0.232], [0.714, 0.346], [0.483, 0.312], [0.478, 0.437],
    [0.525, 0.369],
19     [0.751, 0.489], [0.532, 0.472], [0.473, 0.376], [0.725, 0.445],
    [0.446, 0.459]])
20
21 Train_Size = Train_data.shape[0]
22 mu = rand_row(Train_data, k)
23 belong = np.zeros(Train_Size, dtype=int)
24 size = np.zeros(k, dtype=int)
25
26 while True:
27     flag = False
28     size = np.zeros(k)
29     for i in range(Train_Size):
30         Last_Belong = belong[i]
31         minDis = 0x7fffffff
32         for j in range(k):
33             dis = np.linalg.norm(Train_data[i] - mu[j])
34             if dis < minDis:
35                 belong[i] = j
36                 minDis = dis
37             if belong[i] != Last_Belong:
38                 flag = True
39         size[belong[i]] += 1
40     if not flag:
41         break
42     mu = [0.0, 0.0] * k
43     for i in range(Train_Size):
44         mu[belong[i]] += Train_data[i] / size[belong[i]]
45
46 C = [[] for i in range(k)]
47 for i in range(Train_Size):
48     C[belong[i]].append(Train_data[i])
49
50 for i in range(k):
51     for x in C[i]:
52         print(x)
53     print('\n')

```

[ 0.634	0.264]	我的云
[ 0.608	0.318]	共享文
[ 0.556	0.215]	此电脑
[ 0.666	0.091]	3D 对
[ 0.639	0.161]	视频
[ 0.657	0.198]	图片
[ 0.593	0.042]	文档
[ 0.719	0.103]	下载
[ 0.748	0.232]	音乐
[ 0.483	0.312]	桌面
[ 0.478	0.437]	Wind
[ 0.525	0.369]	DATA
[ 0.473	0.376]	DATA
[ 0.446	0.459]	软件 (
[ 0.697	0.46 ]	网络
[ 0.774	0.376]	
[ 0.714	0.346]	
[ 0.751	0.489]	
[ 0.532	0.472]	
[ 0.725	0.445]	
[ 0.403	0.237]	
[ 0.481	0.149]	
[ 0.437	0.211]	
[ 0.243	0.267]	
[ 0.245	0.057]	
[ 0.343	0.099]	
[ 0.36	0.37]	
[ 0.359	0.188]	项目
[ 0.339	0.241]	
[ 0.282	0.257]	

kmeans可以确认分为3个聚类, 而DBSCAN不可, DBSCAN可以辨别出异常点, 而kmeans不行