# 模式识别实验报告

## 实验一 K-均值聚类

学院：

姓名：

学号：

1. **实验内容**
2. 使用python或Matlab编程实现K-均值聚类算法：要求独立完成算法编程，禁止调用已有函数库或工具箱中的函数；
3. 使用仿真数据测试算法的正确性：将下列19个样本聚成2个聚类：



1. MNIST数据集测试：ClusterSamples中的10000个784维特征手写数字样本聚类为10个类别，根据SampleLabels中的标签统计每个聚类中不同样本的数量。测试不同初始值对聚类结果的影响。
2. **程序代码**

import matplotlib.pyplot as plt

import numpy as np

#计算距离

def distEclud(vecA, vecB):

return np.sqrt(np.sum(np.power(vecA - vecB, 2)))

#得到k个随机质心

def randCent(dataSet, k):

n = np.shape(dataSet)[1]

centroids = np.mat(np.zeros((k, n)))

for j in range(n):

minJ = np.min(dataSet[:, j])

maxJ = np.max(dataSet[:, j])

rangeJ = float(maxJ - minJ)

centroids[:, j] = minJ + rangeJ \* np.random.rand(k, 1)

return centroids

"""

函数：k-均值算法

入口参数：dataSet: 参与聚类的数据集

k: 聚类个数

disMeas: 计算数据集中的点与质心之间的距离

createCent:得到k个随机质心

返回参数：centroids: 聚类完成以后的质心

clusterAssment: 聚类结果，包括 1.某点所属类别 2.与质心的距离的平方(目前好像没用)

"""

def KMeans(dataSet, k, distMeas=distEclud, createCent=randCent):

m = np.shape(dataSet)[0]

clusterAssment = np.mat(np.zeros((m, 2)))

centroids = createCent(dataSet, k)

clusterChanged = True

while clusterChanged:

clusterChanged = False

for i in range(m):

minDist = float('inf')

minIndex = -1

for j in range(k):

distJI = distMeas(centroids[j, :], dataSet[i, :])

if distJI < minDist:

minDist = distJI

minIndex = j

if clusterAssment[i, 0] != minIndex:

clusterChanged = True

clusterAssment[i, :] = minIndex, minDist\*\*2

for cent in range(k):

ptsInClust = dataSet[np.nonzero(clusterAssment[:, 0].A == cent)[0]]

centroids[cent, :] = np.mean(ptsInClust, axis=0)

return centroids, clusterAssment

#画出聚类结果

def plotDataSet(dataSet):

Centroids, clusterAssment = KMeans(dataSet, 2)

dataSet = dataSet.tolist()

Centroids = Centroids.tolist()

clusterAssment = clusterAssment.tolist()

xcord = [[], []]

ycord = [[], []]

m = len(clusterAssment)

for i in range(m):

if int(clusterAssment[i][0]) == 0:

xcord[0].append(dataSet[i][0])

ycord[0].append(dataSet[i][1])

elif int(clusterAssment[i][0]) == 1:

xcord[1].append(dataSet[i][0])

ycord[1].append(dataSet[i][1])

fig = plt.figure()

ax = fig.add\_subplot(111)

ax.scatter(xcord[0], ycord[0], s=20, c='b', marker='\*', alpha=.5)

ax.scatter(xcord[1], ycord[1], s=20, c='r', marker='o', alpha=.5)

ax.scatter(Centroids[0][0], Centroids[0][1], s=100, c='k', marker='+', alpha=.5)

ax.scatter(Centroids[1][0], Centroids[1][1], s=100, c='k', marker='+', alpha=.5)

plt.title('DataSet')

plt.xlabel('X')

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

data = [[0, 0], [1, 0], [0, 1], [1, 1],

[2, 1], [1, 2], [2, 2], [3, 2],

[6, 6], [7, 6], [8, 6], [7, 7],

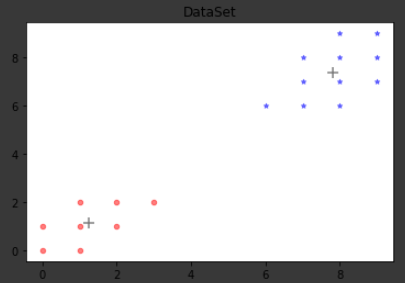
[8, 7], [9, 7], [7, 8], [8, 8],

[9, 8], [8, 9], [9, 9]]

dataSet = np.mat(data)

plotDataSet(dataSet)

1. **实验结果**
2. 仿真数据实验结果：（可以列出每个聚类中包含的样本，也可以画图显示不同聚类）



1. MNIST数据集实验结果：

**每个聚类中包含不同类别样本数量统计表**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 聚类0 |  |  |  |  |  |  |  |  |  |  |
| 聚类1 |  |  |  |  |  |  |  |  |  |  |
| 聚类2 |  |  |  |  |  |  |  |  |  |  |
| 聚类3 |  |  |  |  |  |  |  |  |  |  |
| 聚类4 |  |  |  |  |  |  |  |  |  |  |
| 聚类5 |  |  |  |  |  |  |  |  |  |  |
| 聚类6 |  |  |  |  |  |  |  |  |  |  |
| 聚类7 |  |  |  |  |  |  |  |  |  |  |
| 聚类8 |  |  |  |  |  |  |  |  |  |  |
| 聚类9 |  |  |  |  |  |  |  |  |  |  |