

例 5.7 Python 程序代码

```
# dataSet 样本点, k 簇的个数
# disMeas 距离量度, 默认为欧几里得距离
# createCent, 初始点的选取
def kMeans(dataSet, k, distMeas=distEclud, createCent=randCent):
    m = shape(dataSet)[0] #样本数
    clusterAssment = mat(zeros((m, 2))) #m*2 的矩阵
    centroids = createCent(dataSet, k) #初始化 k 个中心
    clusterChanged = True
    while clusterChanged: #当聚类不再变化
        clusterChanged = False
        for i in range(m):
            minDist = inf; minIndex = -1
            for j in range(k): #找到最近的质心
                distJI = distMeas(centroids[j, :], dataSet[i, :])
                if distJI < minDist:
                    minDist = distJI; minIndex = j
            if clusterAssment[i, 0] != minIndex: clusterChanged = True
            # 第 1 列为所属质心, 第 2 列为距离
            clusterAssment[i, :] = minIndex, minDist**2
        print centroids

    # 更改质心位置
    for cent in range(k):
        ptsInClust = dataSet[nonzero(clusterAssment[:, 0].A==cent)[0]]
        centroids[cent, :] = mean(ptsInClust, axis=0)
    return centroids, clusterAssment

def loadDataSet(fileName): #general function to parse tab -delimited floats
    dataMat = [] #assume last column is target value
    fr = open(fileName)
    for line in fr.readlines():
        curLine = line.strip().split('\t')
        fltLine = map(float, curLine) #map all elements to float()
        dataMat.append(fltLine)
    return dataMat

def distEclud(vecA, vecB):
    return sqrt(sum(power(vecA - vecB, 2))) #1a. norm(vecA-vecB)

def randCent(dataSet, k):
    n = shape(dataSet)[1]
    centroids = mat(zeros((k, n))) #create centroid mat
    for j in range(n): #create random cluster centers, within bounds of each dimension
        minJ = min(dataSet[:, j])
```

```

        rangeJ = float(max(dataSet[:, j]) - minJ)
        centroids[:, j] = mat(minJ + rangeJ * random.rand(k, 1))
    return centroids
#绘图的代码如下
def draw(data, center):
    length=len(center)
    fig=plt.figure
    # 绘制原始数据的散点图
    plt.scatter(data[:, 0], data[:, 1], s=25, alpha=0.4)
    # 绘制簇的质心点
    for i in range(length):
        plt.annotate('center', xy=(center[i, 0], center[i, 1]), xytext=\
            (center[i, 0]+1, center[i, 1]+1), arrowprops=dict(facecolor='red'))
    plt.show()#绘图的代码如下

```