电子商务导论

第四章 商务数据挖掘

朱桂祥 (9120201070@nufe.edu.cn)
南京财经大学信息工程学院
江苏省电子商务重点实验室
电子商务信息处理国家级国际联合研究中心
电子商务交易技术国家地方联合工程实验室

清华大学校训

---易经乾卦《象》曰：天行健，君子以自强不息。

---易经坤卦《象》曰：地势坤，君子以厚德载物。

君子自励犹如天体之运行刚健不息，不得一曝十寒，不应见利而进，知难而退，而应重自胜摈私欲尚果毅，不屈不挠，见义勇为，不避艰险，自强不息；同时，君子应如大地的气势厚实和顺，容载万物，责己严，责人轻，以博大之襟怀，吸收新文明，改良我社会，促进我政治，以宽厚的道德，担负起历史重任。

易经乾卦解读

九三：君子终日乾乾，夕惕若，厉无咎

[illegible]

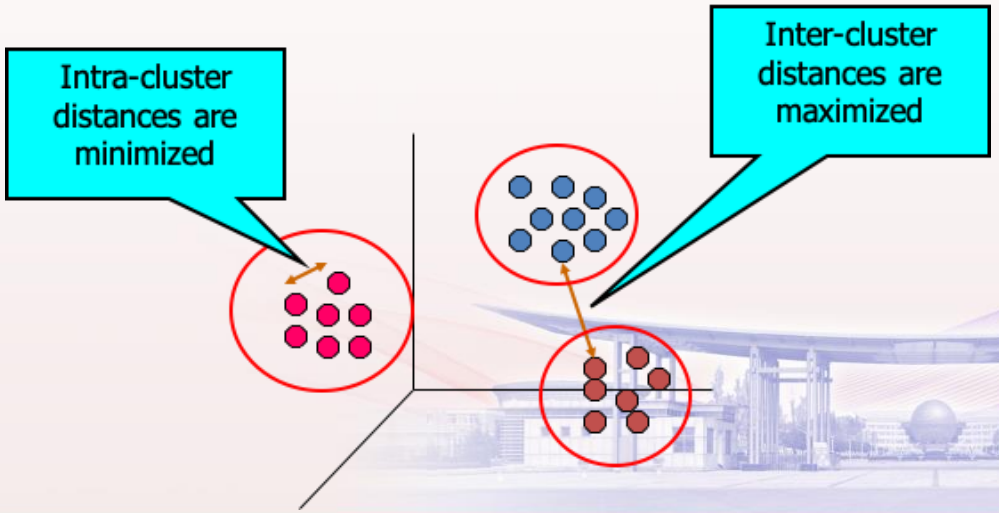
目 录
Contents

三、聚类

1. 背景

聚类分析的目的是对数据进行分组，使得分到同一组中的数据更为相似，不同组的数据差异较大

- 分类是监督学习 (supervised learning)，有训练集
- 聚类是无监督学习 (unsupervised learning)，注意简单划分不是聚类（如按班级或性别把学生划分不同的组）



三、聚类

2. 聚类算法的应用

✓零售业

将经常同时购买的数据项聚类到一起有利于改善商品的布置，提高销售利润。

将具有相似的购买模式的顾客聚类到一起，分析每一类顾客的特征，有利于对特定的顾客群进行特定商品的宣传和销售

✓信息检索

对文档进行分类，改善信息检索的效率，或者发现某一领域文献的组成结构

✓医疗分析

对一组新型疾病聚类，得到每类疾病的特征描述，对这些疾病进行识别，提高治疗的功效

发现不属于正常类别的特殊病例，识别组织的病变细胞

✓天文学

利用聚类分析宇宙仿真系统得到的数据，更好地理解黑洞形成和进化的物理过程

三、聚类

2.聚类算法的应用

例子：

- ✓ 市场部想提高客户满意度和客户保有率，计划实行创办《每周赠券》杂志，将杂志送给客户群，以鼓励他们访问 FoodMart 商店。为了定义《每周赠券》杂志，市场部想将客户群划分为三个类别。根据三个组的特征，市场部可以选择赠券的类型，以便插入各个版本的《每周赠券》杂志
- ✓ 选择想要在算法中表示各个客户类别特性的人口统计特征列表：婚姻状况、年收入、在家子女数、教育程度…… 然后训练此模型，最终使其能够浏览受训数据并从中分析三种客户类别
- ✓ 市场部将根据每个客户类别的人口统计属性，选择将要插入《每周赠券》杂志各个版本中的赠券列表

三、聚类

3. 经典的聚类算法

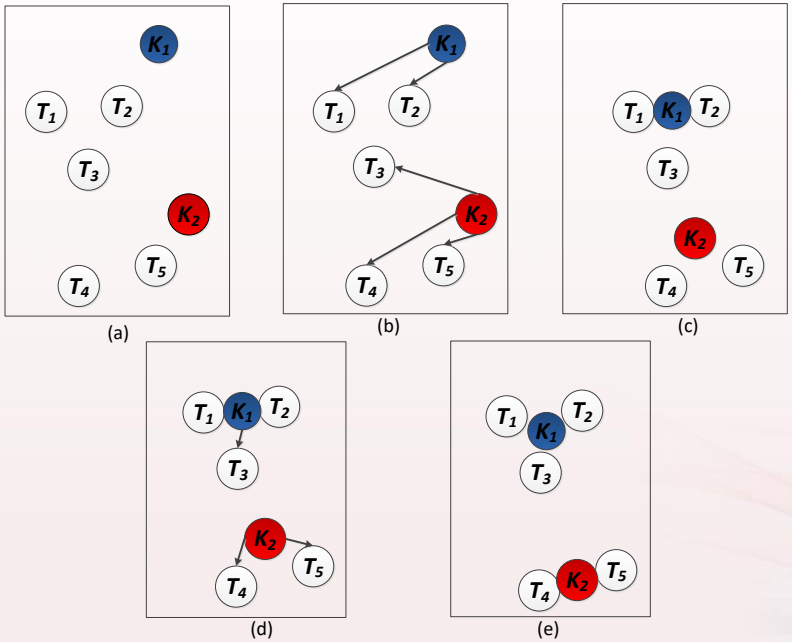
- ✓ 基于原型：各簇可以用概率分布或者中心点刻画
  - – K-means、混合模型 (mixture model, MM)、模糊C-均值 (Fuzzy C-means, FCM)、自组织映射 (self-organizing maps, SOM)…
- ✓ 基于密度：高密度区域（各簇）被低密度区域分开
  - – DBSCAN、CLIQUE、DENCLUE…
- ✓ 基于图：数据为顶点、距离为边，凝聚顶点或划分边
  - – 凝聚层次法 (agglomerative hierarchical clustering, AHC)、Jarvis-Patrick Clustering (JP)、谱聚类(如MinMaxCut)…
- ✓ 混合方法：聚类算法之间或者与其他方法的组合

  一致性聚类(consensus clustering)

三、聚类

4. K均值(K-means)聚类

K均值算法思想有直观的几何意义：将样本点聚集（归属）到距离它最近的那个聚类中心。找出数据集中的K个聚类中心是算法的目标（简单起见，这里使用欧式距离来度量样本间的相似度）。



基本K均值算法是一个两步的迭代过程：

Algorithm 1 Basic K-means Algorithm.

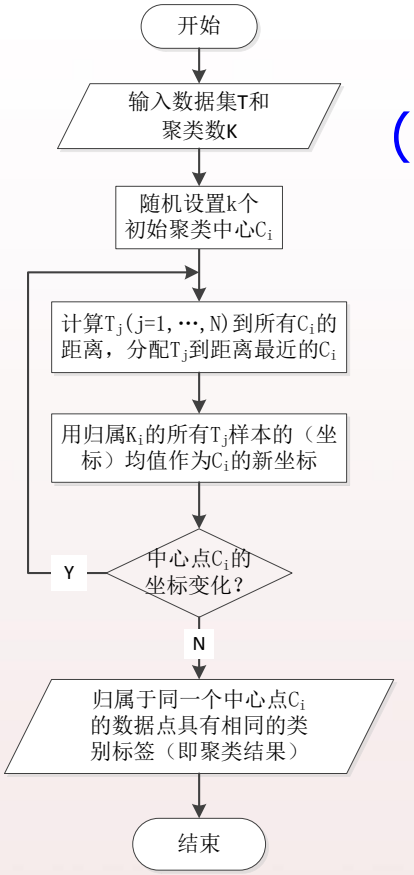
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

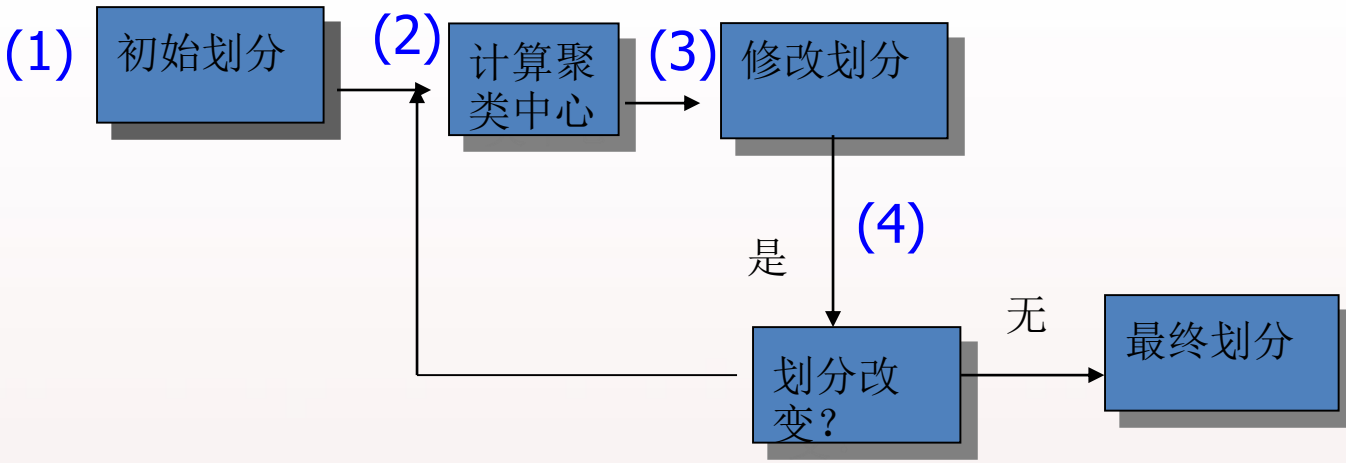
将数据分配到最近的中心

根据分配的数据重新计算中心

三、聚类

4. K均值(K-means)聚类





三、聚类

4. K均值(K-means)聚类

K均值(K-means) 算法Python实现：

(1) initCentroids函数根据当前样本数据集和指定的k，随机生成k个中心点，用于聚类。

```
# 随机生成聚类中心点
#参数：dataSet-List列表，已分类点坐标
#      k-整数，近邻数量
#返回值：centroids-2维列表，k个随机中心点坐标
def initCentroids(dataSet, k):
    numSamples, dim = dataSet.shape
    centroids = np.zeros((k, dim))
    for i in range(k):
        index = int(np.random.uniform(0, numSamples))
        centroids[i, :] = dataSet[index, :]
    return centroids
```

三、聚类

4. K均值(K-r

K均值(K-means)

(2) K均值聚类：

```
# K均值聚类
#参数：dataSet-List列表，待聚类样本集
#      k-整数，近邻数量
#返回值：centroids-2维列表，k个随机中心点坐标
#       clusterAssment -列表，各个样本点的聚类结果
def kmeans(dataSet, k):
    numSamples = dataSet.shape[0]
#第一列数据存放归属的点
#第二列存放样本与候选聚类中心点之间的误差
    clusterAssment = np.mat(np.zeros((numSamples, 2)))
    clusterChanged = True
    centroids = initCentroids(dataSet, k)

    while clusterChanged:
        clusterChanged = False
        for i in range(numSamples):
            minDist  = 100000.0
            minIndex = 0
#依次找出最近候选聚类中心点
            for j in range(k):
                distance = euclDistance(centroids[j, :], dataSet[i, :])
                if distance < minDist:
                    minDist  = distance
                    minIndex = j
#更新归属结果
            if clusterAssment[i, 0] != minIndex:
                clusterChanged = True
                clusterAssment[i, :] = minIndex, minDist**2
#更新候选聚类中心点坐标
        for j in range(k):
            pointsInCluster = dataSet[np.nonzero(clusterAssment[:, 0].A == j)[0]]
            centroids[j, :] = np.mean(pointsInCluster, axis = 0)

    print('KMN聚类完成!')
    return centroids, clusterAssment
```

三、聚

4. K均值

K均值(K-

(3) 2维平面

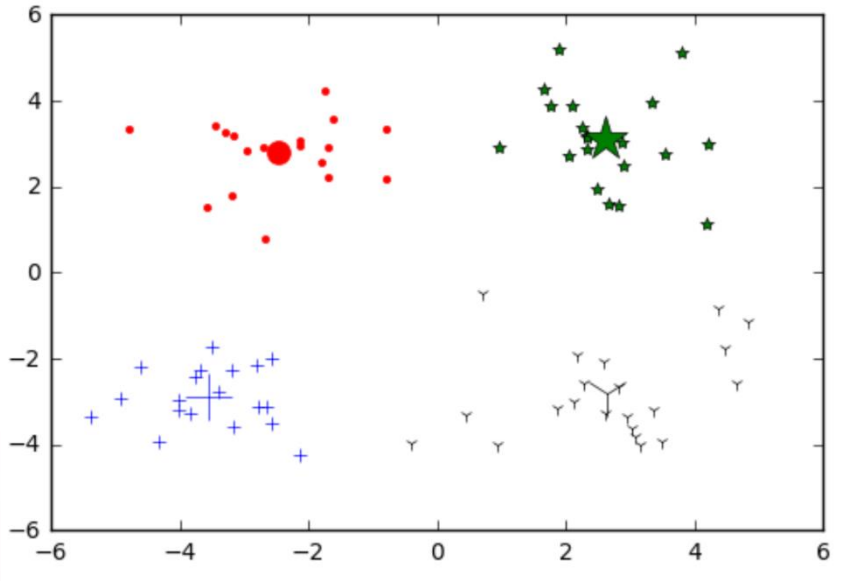
```
# 2维平面显示聚类结果
#参数：dataSet-List列表，样本集
#      k-整数，近邻数量
#      centroids-List列表，聚类中心点坐标
#      clusterAssment-List列表，聚类结果
#返回值：无
def showCluster(dataSet, k, centroids, clusterAssment):
    fig_2d_clustered=plt.figure()
    ax2d_clustered=fig_2d_clustered.add_subplot(111)

    numSamples, dim = dataSet.shape
    if dim != 2:
        print("只能绘制2维图形")
        return 1
#创建数据点标记格式控制列表，实现数据点区别输出
    mark = ['.r', '+b', '*g', '1k', '^r', 'vr', 'sr', 'dr', '<r', 'pr']
    if k > len(mark):
        print("K值过大！")
        return 1
#绘制所有样本点
    for i in range(numSamples):
        markIndex = int(clusterAssment[i, 0])
        ax2d_clustered.plot(dataSet[i, 0], dataSet[i, 1], mark[markIndex])

#绘制聚类中心点
    for i in range(k):
        ax2d_clustered.plot(centroids[i, 0], centroids[i, 1], mark[i], markersize = 20)

    fig_2d_clustered.savefig('clusterRes.png', dpi=300, bbox_inches='tight')
    fig_2d_clustered.show()
```

三、聚类



4. K均值(K-means)聚类

K均值(K-means) 算法Python实现：

(4)调用以上函数，对读入数据进行聚类：

```
#调用以上函数，对读入数据进行聚类
print("step 1: 读入数据：")
dataSetKMN = []
fileIn = open('testSet.txt')
for line in fileIn.readlines():
    lineArr = line.strip().split(' ')
    dataSetKMN.append([float(lineArr[0]), float(lineArr[1])])
dataSetKMNSize = len(dataSetKMN)
dataSetKMN = np.mat(dataSetKMN)
for i in range(dataSetKMNSize):
    plt.plot(dataSetKMN[i, 0], dataSetKMN[i, 1],'b*')
print("原始数据分布：")
plt.savefig('4_3:_kmn_orig.png', dpi=300, bbox_inches='tight')
plt.show()
```

三、聚类

4. K均值(K-means)聚类

K均值(K-means) 算法Python实现：

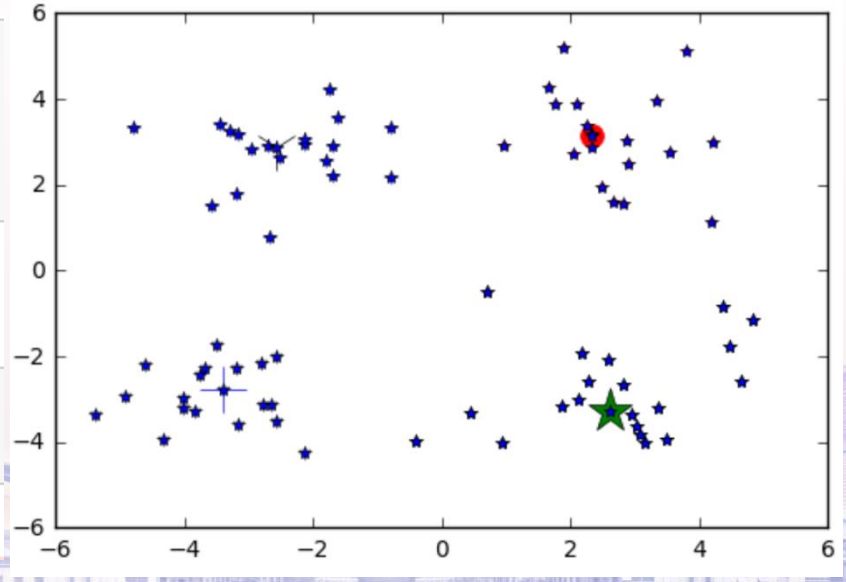
(5)根据聚类结果用不同的样式显示不同聚类的数据点，并且突出显示了算法求出的k=4个聚类中心点。

```
#K取值4，调用K均值算法聚类
print("step 2: 聚类")
k = 4
centroids, clusterAssment = kmeans(dataSetKMN, k)
```

```
step 2: 聚类
KMN聚类完成!
```

```
print("step 3: 结果输出：")
showCluster(dataSetKMN, k, centroids, clusterAssment)
```

```
step 3: 结果输出：
```



谢谢观赏 下节课见