k-means聚类算法

K-Means Clustering Al

成都信息工程大学学生数学建模协会

https://blog.csdn.net/zzzzzzzxxaaa/article/details/126742111

https://blog.csdn.net/sikh_0529/article/details/126806720

https://www.bilibili.com/video/BV1py4y1r7DN/?spm_id_from=333.337.search-card.all.click

主讲人：gym

目录

CONTENTS

01 / 算法原理

Principle of the algorithm

什么是K-means聚类算法呢

k均值聚类算法（k-means clustering algorithm）是一种迭代求解的聚类分析算法，其步骤是，预将数据分为K组，则随机选取K个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。每分配一个样本，聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有（或最小数目）对象被重新分配给不同的聚类，没有（或最小数目）聚类中心再发生变化，误差平方和局部最小。

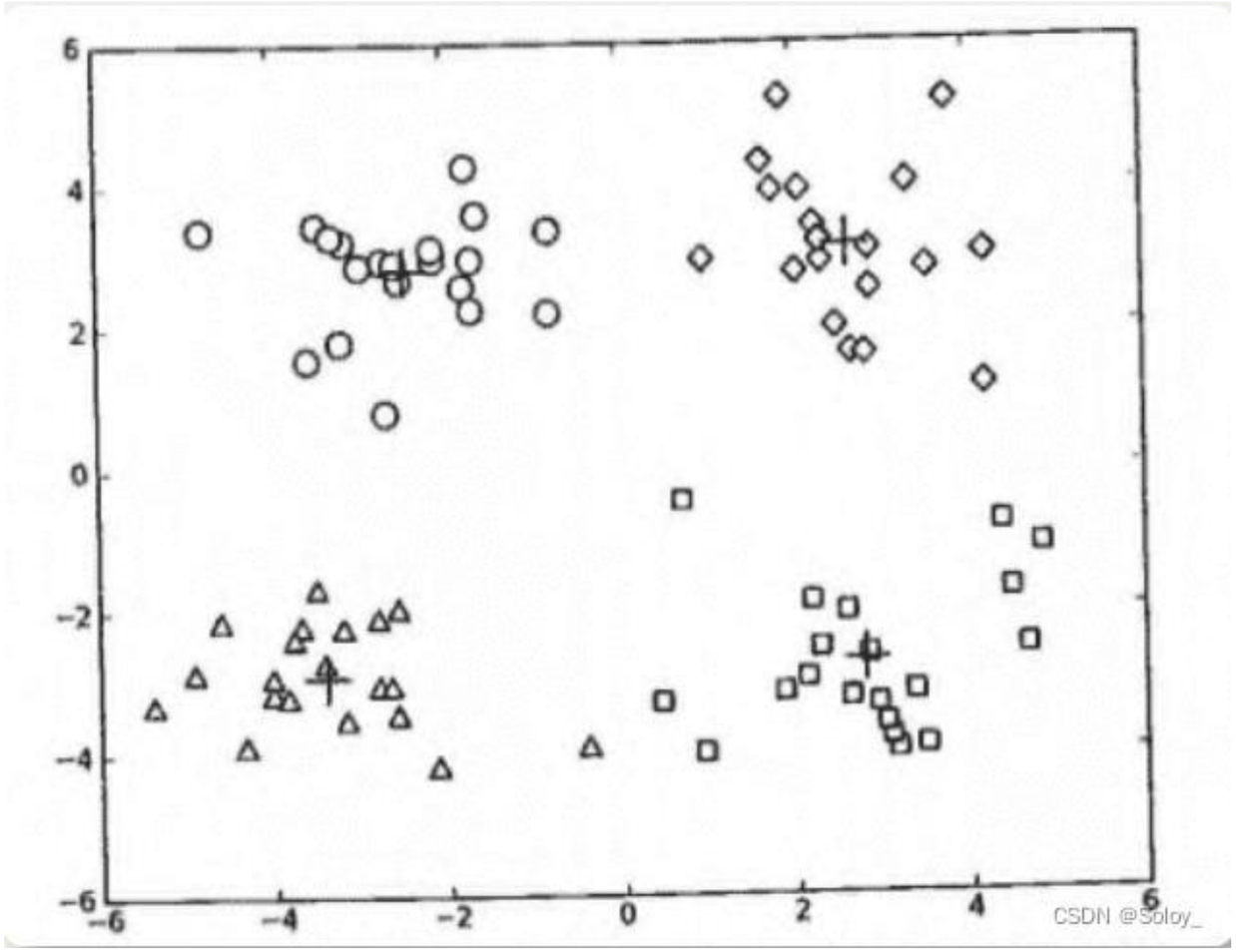
K-means聚类算法

1.类型：其属于无监督学习算法，即从无标签数据中寻找隐藏的规律
2.目标：将n个观测数据点按照一定的标准划分为k个聚类，每个聚类数据（观测点）相似（簇）
3.实现过程：（1）：根据n个数据确定k个点作为初始质心
（2）：为每个数据点找到最近的质心，并分配给该簇
（3）：重新计算每个簇的质心，也就是平均值
（4）：重复（2）和（3）直到所有点距离最近的质心为其对应的质心

02 / 简单例题

Example questions

从生活中理解



我们假设这些是一个地区的居民布，我们要开几家餐馆

题目：有一下6个点，将B和E点作为两个簇的初始簇心，问最后簇的归属情况。

	X	Y
A	1	3
B	2	4
C	3	1
D	3	6
E	5	2
F	5	5

	X	Y
M1	2	4.3
M2	4.3	2.7

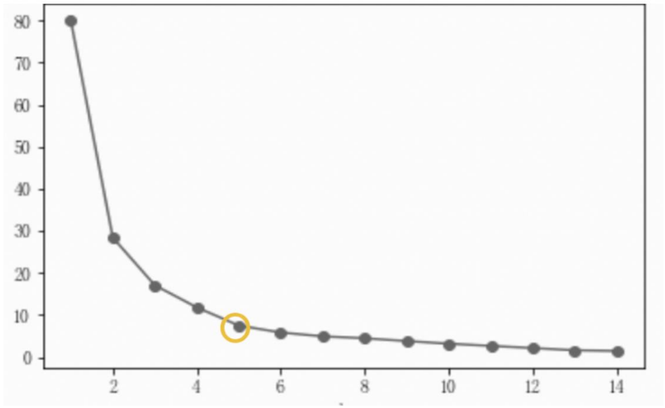
	X	Y	d1	d2
A	1	3	1.41	4.12
B	2	4	0	3.61
C	3	1	3.16	2.24
D	3	6	2.24	4.47
E	5	2	3.61	0
F	5	5	3.16	2.24

	X	Y	d1	d2
A	1	3	1.64	3.31
B	2	4	0.30	2.64
C	3	1	3.45	2.14
D	3	6	1.97	3.54
E	5	2	3.78	0.99
F	5	5	3.08	2.40

03 / 代码概述

Code Outlining

1.首先呢就是在读取数据和选定特定的特征（一定情况下的分类方式）之后确定K个初始点作为质心。



肘部法则确定K值，纵坐标是SSE，也就是K的每一个取值（横坐标）所对应的误差平方和，当曲线趋于缓和时的拐角取其为K值。

2.距离计算选用欧几里得距离：

$$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2}$$

```
def kMeans(dataMat, k, distMeas=distEclud, createCent=randCent):
    m = shape(dataMat)[0]  # 行数
    clusterAssment = mat(zeros(
        (m, 2)))  # 创建一个与 dataMat 行数一样，但是有两列的矩阵，用来保存簇分配结果
    centroids = createCent(dataMat, k)  # 创建质心，随机k个质心
    clusterChanged = True
    while clusterChanged:
        clusterChanged = False
        for i in range(m):  # 循环每一个数据点并分配到最近的质心中去
            minDist = inf
            minIndex = -1
            for j in range(k):
                distJI = distMeas(centroids[j, :],
                                  dataMat[i, :])  # 计算数据点到质心的距离
                if distJI < minDist:  # 如果距离比 minDist（最小距离）还小，更新 minDist（最小距离）和最小质心的 index（索引）
                    minDist = distJI
                    minIndex = j
            if clusterAssment[i, 0] != minIndex:  # 簇分配结果改变
                clusterChanged = True  # 簇改变
                clusterAssment[
                    i, :] = minIndex, minDist**2  # 更新簇分配结果为最小质心的 index（索引），minDist（最小距离）的平方
        print(centroids)
        for cent in range(k):  # 更新质心
            ptsInClust = dataMat[nonzero(
                clusterAssment[:, 0].A == cent)[0]]  # 获取该簇中的所有点
            centroids[cent, :] = mean(
                ptsInClust, axis=0)  # 将质心修改为簇中所有点的平均值，mean 就是求平均值的
    return centroids, clusterAssment
```

04 / 应用与局限性

Applications and limitations

感谢聆听

现实世界的奥秘等你探索和发现，

体验数学魅力， 让你收益终身！