**实验KMeans**

姓名: 李昀哲 学号： 20123101

**1 题目和数据：**

题目：K-means

通过对给定数据进行聚类分析来了解K-means算法。对K-means算法原理，K-means算法流程，K-means算法应用有更深的了解

数据描述:

本数据是随机生成的符合高斯分布的二维样本点。通过查看数据的shape，观察到其为80行2列，即80组样本，每组样本有两个特征。数据点的分布如图1所示

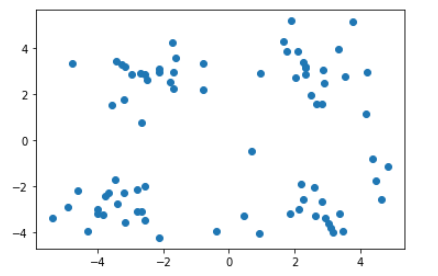


图1 K-means实验数据

**2 算法：**

KMeans聚类是最基础常用的聚类算法，聚类属于非监督学习。它的基本思想是，通过迭代寻找K个簇（Cluster）的一种划分方案，使得聚类结果对应的损失函数最小。其中，损失函数可以定义为各个样本距离所属簇中心点的误差平方和 *J*.

与分类、序列标注等任务不同，聚类是在事先并不知道任何样本标签的情况下，通过数据之间的内在关系把样本划分为若干类别，使得同类别样本之间的相似度高，不同类别之间的样本相似度低（即增大类内聚，减少类间距）。

KMeans最核心的部分就是先固定中心点，调整每个样本所属的类别来减少*J*；再固定每个样本的类别，调整中心点继续减小*J*。两个过程交替循环，*J*单调递减直到最（极）小值，中心点和样本划分的类别同时收敛。

其**优点**是算法思想简单，便于实现；但**缺点**也较为明显：K值难以确定，复杂程度与样本呈线性关系，很难发现任意形状的cluster，如图2所示。因此往往也会通过DBSCAN算法处理。

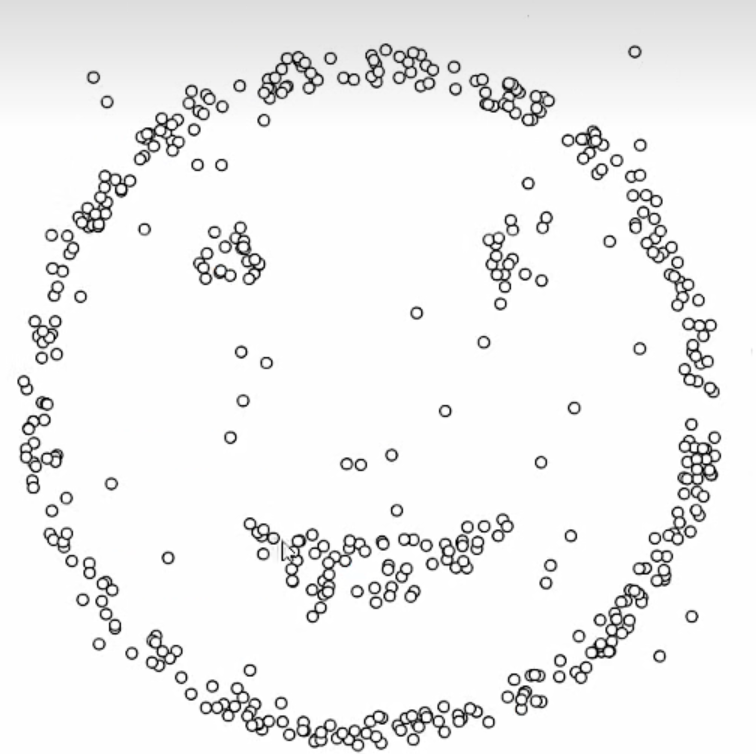
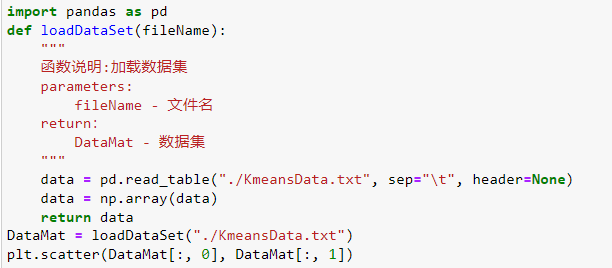


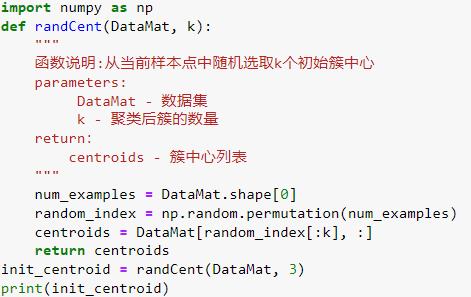
图2 K-means难以处理的形状

**3 代码及结果**

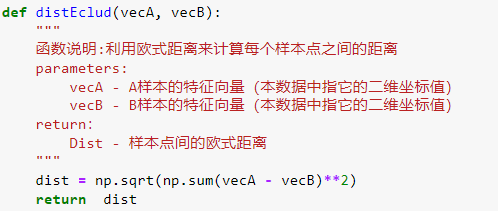
首先加载数据集，得到DataMat并进行展示；



再根据算法流程，找到初始的随机簇中心；



得到初始簇中心后需要计算每个点到每个中心的距离，以进行迭代，因此需要用到计算两个点之间的欧几里得距离；



所有算法工具函数准备齐全后，进行整体的迭代，其中需要注意的是，在得到最近簇中心的过程中，要记录的并非到簇中心的对短距离，而是记录这个簇中心的索引值，此处在测试是发生错误，多次调试后才发现。

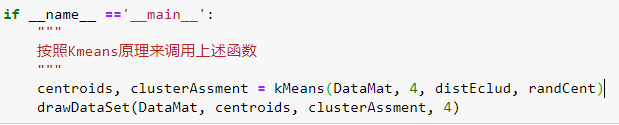
另外，对于numpy.array而言，用一个数去 == numpy.array，会得到一个该索引处是否存在的bool数组，可以用于作为另一个numpy.array的索引。



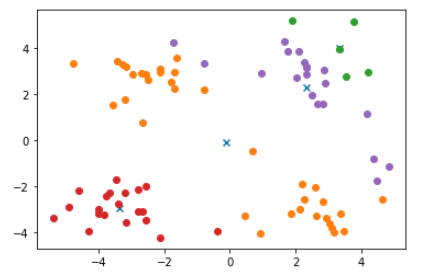
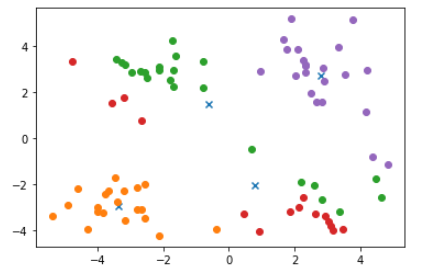
最后展示结果



主函数如下：



由初始数据点可以观察到，大概是会分为4簇，因此指定k为4，但对于迭代得到的结果，并不如预先所想那样分配，多组实验结果都以斜向划分，猜测可能与初始点的选择有关，且原数据并非十分规整，导致从人的角度观察，也并不能完美划分各个簇，因此认为这些结果是合理的。



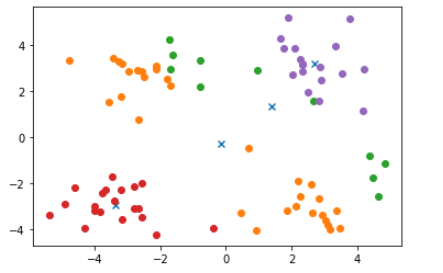
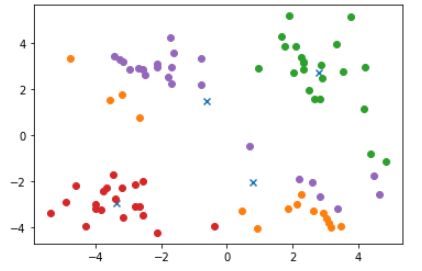


图3 实验结果