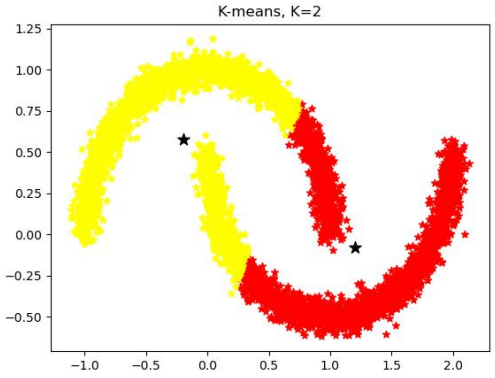
系统工程导论作业六——聚类分析

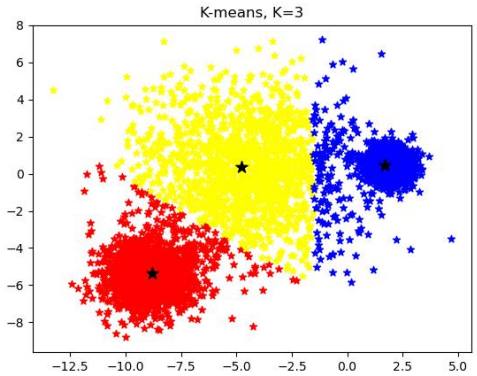
彭程 2020011075

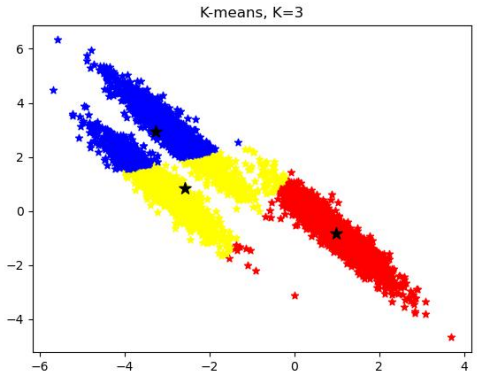
1.K-means

1.1 自行编写 K-means 聚类算法，绘制 3 个数据集的聚类结果

根据三个聚类的形状，分别选择 K 值为 2，3，3，得到的聚类结果如下：







1.2. 利用数据集 data2, 对 K-means 算法进行如下实验

1.2.1 增加聚类数目，计算并分析聚类结果，决定最合适的聚类数目并说明理由

$k = 1 \sim 9$ 的聚类结果如下：

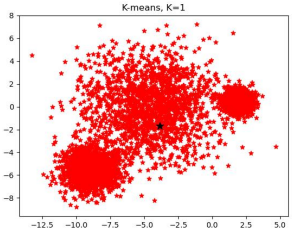


图 1: k=1

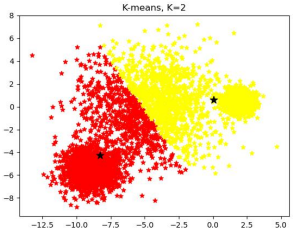


图 2: k=2

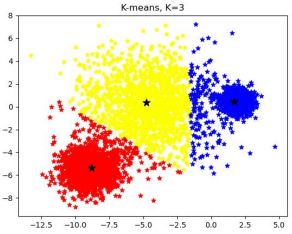


图 3: k=3

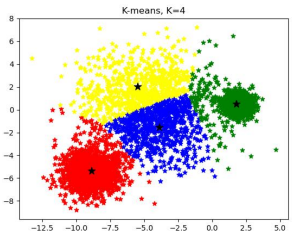


图 4: k=4

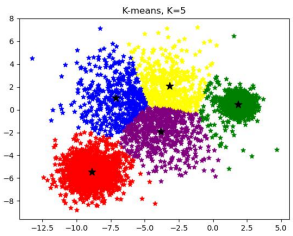


图 5: k=5

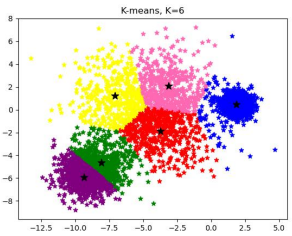


图 6: k=6

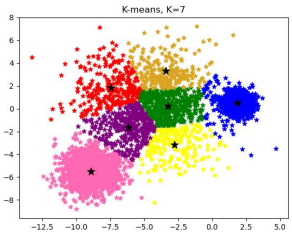


图 7: k=7

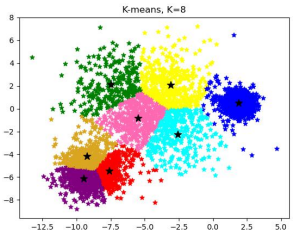


图 8: k=8

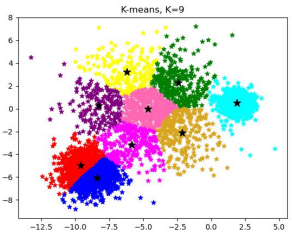
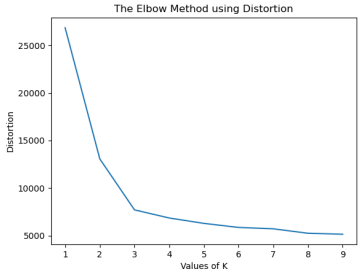


图 9: k=9

采用 elbow method 法则选取 K, 即最小化点到聚类中心的距离之和:

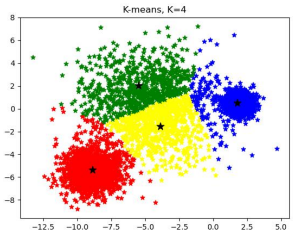
$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right)$$

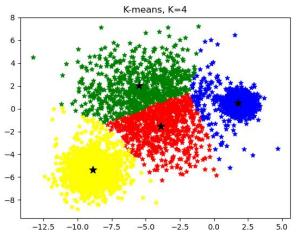
绘制出的图像如下：

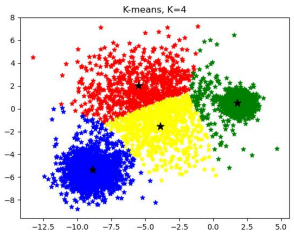


故选择 k = 3。因为 k>3 后随 k 的增加，距离和的变换趋于平缓，增加 k 的代价过高。

1.2.2 选择不同的初始点多次实验，观察初始点的选择对最终结果的影响，并分析原因

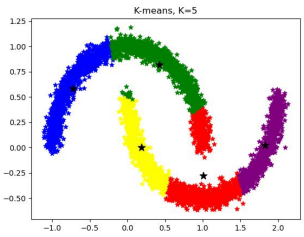


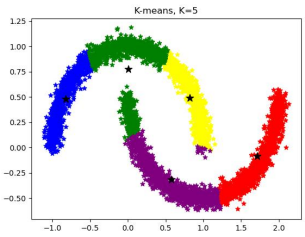


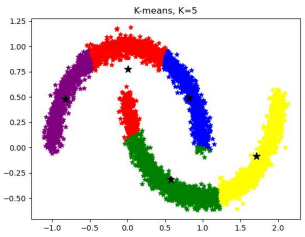


选择不同的初始点进行多次实验，发现最终都收敛到了相同的中心点（如上），只是由于初始点不同，在收敛时间上有差异。

这可能是因为 data2 自身的点分布比较分散且均匀，于是我们对 data1 进行了实验，结果如下：





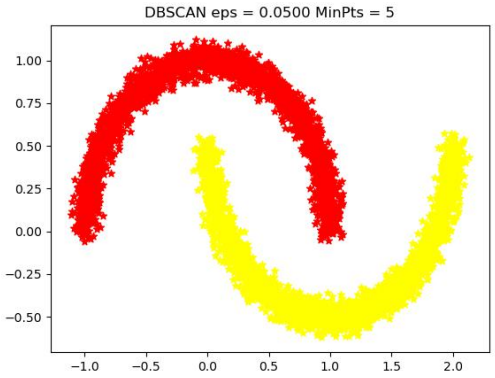


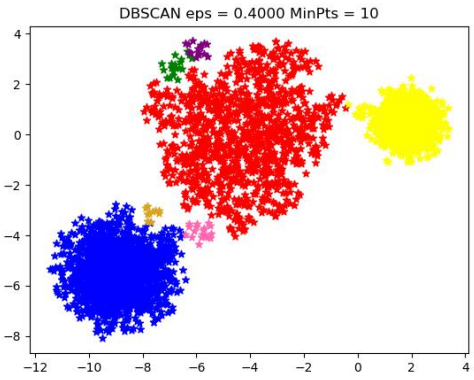
对于 data1 的实验我们得到了不同的结果，这是因为 K-means 算法初始点选取不同时，虽然能保证收敛到某一个结果，但不能保证每次收敛到同一个结果，最后的结果可能是不稳定的，也可以理解为陷入局部的极值。

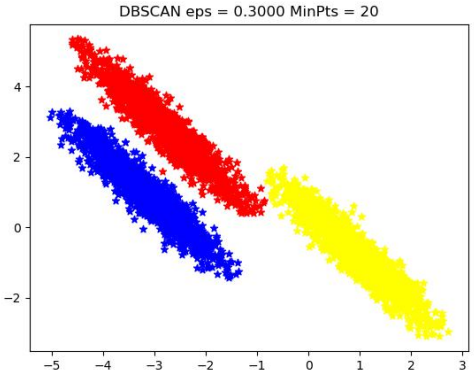
2.DBSCAN

2.1 自行编写 DBSCAN 聚类算法，绘制 3 个数据集的聚类结果

得到的聚类结果如下：







2.2 利用数据集 data3, 对 DBSCAN 算法进行如下实验

2.2.1 选择不同的 ϵ, 观察实验结果并分析原因;

ϵ 分别取 0.01,0.1,0.2,0.3,0.4,0.5 的聚类结果如下所示:

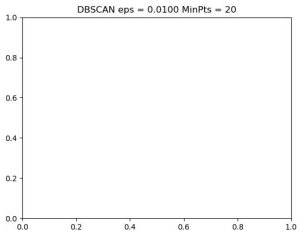


图 10: $\epsilon = 0.01$

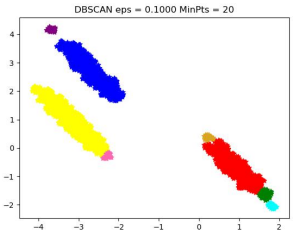


图 11: $\epsilon = 0.1$

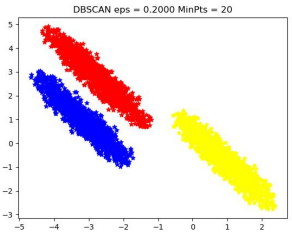


图 12: $\epsilon = 0.2$

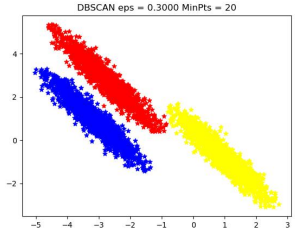


图 13: $\epsilon = 0.3$

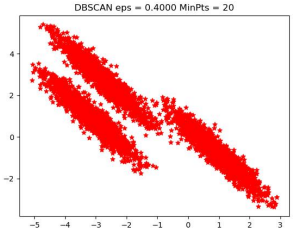


图 14: $\epsilon = 0.4$

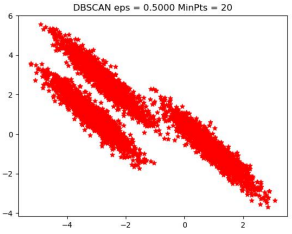


图 15: $\epsilon = 0.5$

$\epsilon = 0.01$ 时半径选取过小，导致不存在核心点。$\epsilon = 0.1$ 时半径仍然较小，导致把同一类的样本点拆分成了几类。$\epsilon = 0.2, 0.3$ 时半径合适，分类结果较为可靠。$\epsilon = 0.4, 0.5$ 时半径过大，几个类会被粘连到一起，导致聚类效果变差。

2.2.2 选择不同的 $minPots$，观察实验结果并分析原因；

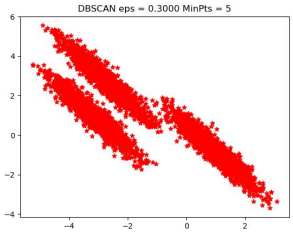


图 16: $minPots = 5$

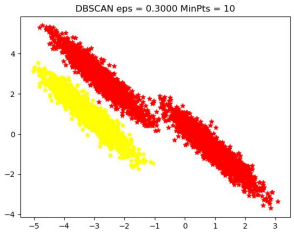


图 17: $minPots = 10$

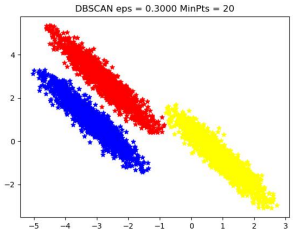


图 18: $minPots = 20$

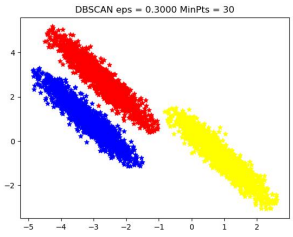


图 19: $minPots = 30$

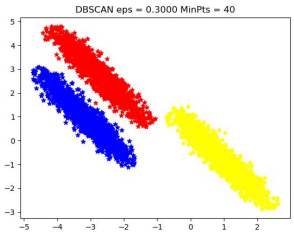


图 20: $minPots = 40$

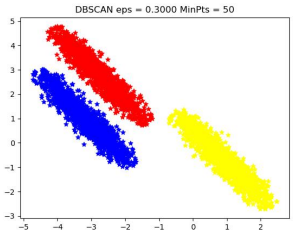


图 21: $minPots = 50$

当 ϵ 一定时，增大 MinPots 即意味着一个点要想成为核心点需要邻域内有更多的点，即使得一个点成为核心点的要求变高，因而会导致部分原本的核心点被当作边界点或者被舍弃的噪声点，会把同一类的样本点拆分成几类，甚至导致不存在核心点；反之减小 MinPots 会使得一个点成为核心点的要求变低，因而会导致更多的边界点或者被舍弃的噪声点会被当作核心点，甚至相距较近的几个类会被粘连到一起，导致聚类效果变差。

3. 对比分析 kmeans 和 DBSCAN 聚类算法

K-means 算法是基于划分的聚类算法，算法简单、快速、易于实现，比较适合球状分布的聚类，而且不受点的密度的影响，聚类的结果易于解释，但是可能会陷入局部最优，对离群点和噪声点敏感，不同初始点的选取可能会导致不同的聚类结果。

DBSCAN 算法是基于密度的聚类算法，可以对任意形状的稠密数据集进行聚类，而且可以在聚类的同时发现噪音点，不需要事先指定类别数目，聚类结果也不依赖节点的遍历顺序，但是数据集过大时，收敛时间长，而且 ϵ、MinPots 选取较为困难。

具体到本次作业中的三组数据，对于 data1 和 data3，DBSCAN 得到的聚类结果更好；对于 data2，K-means 和 DBSCAN 效果相对来说比较接近，但 DBSCAN 的结果出现了许多小聚类，K-means 则不会出现。而且 K-means 收敛时间明显小于 DBSCAN 的收敛时间。