智能数据挖掘作业 7

1920030004 黄铭瑞

实验目的

理解 DBSCAN 聚类算法的原理，并使用 DBSCAN 聚类算法对 2d4c、long、moon、 sizes5、 smile、 spiral、 square1、 square4 数据集进行聚类。

实验原理

DBSCAN 聚类

定义

DBSCAN（Density-Based Spatial Clustering of Applications with Noise，具有噪声的基于密度的聚类方法）是一种基于密度的空间聚类算法。该算法将具有足够密度的区域划分为簇，并在具有噪声的空间数据库中发现任意形状的簇，DBSCAN 算法将“簇”定义为密度相连的点的最大集合。

三类数据点

- **核心点（core point）：**若样本x_i的 epsilon 邻域内至少包含了 MinPts 个样本，则称x_i为核心点。
- **边界点（border point）：**若样本x_i的 epsilon 邻域内包含的样本数目小于 MinPts，但他在其他核心点的邻域内，则称x_i为边界点。
- **噪声点（noise）：**既不是核心点也不是边界点的点。噪声点是不被聚类纳入的点。

密度相关

- **密度直达（directly density reachable）：**如果满足$p \in N_{eps}(q), and \left|N_{eps}(q)\right| \geq MinPts$，那么样本点$p$是由样本点$q$对于参数$\{eps, MinPts\}$密度直达。
- **密度可达（density reachable）：**如果存在一系列样本点，$q \to a \to b \to c \to d \to p$，任意两个相邻对象之间是直接密度可达，则$p$是$q$关于参数$\{eps, MinPts\}$密度可达。

- **密度相连（density connected）：**如果在样本集D中存在一个样本点o，使得p和q均由样本点o密度可达，那么称p与q对于参数$\{eps, MinPts\}$密度相连。

算法步骤

Step1: 任选一个点q，找到和它距离小于等于eps的所有点。如果找到的点数小于$MinPts$，标记q为噪声点；如果点的个数大于$MinPts$，这个点q被标为核心样本，并分配新的簇标签。

Step2: 访问点q的所有邻居点，如果他们还未被分配到一个簇，那么将刚创建的簇标签分配给他们，如果他们是核心点，就依次访问他们的邻居，以此进行簇扩张，直到eps内没有更多核心样本。

Step3: 重复 **step1**、**step2**，直到没有新的簇添加，结束。

DBSCAN 参数选择

eps 选择

eps参数设置过小，大部分数据不能聚类；eps参数设置过大，多个簇会归到一个大簇中。

可以通过绘制 k-distance 曲线的方法寻找eps合适值。给定K邻域参数k，对于数据中的每个点，计算对应的第k个最近邻域距离，并将数据集所有点对应的最近邻域距离进行从大到小排序后绘图，称这幅图为排序的k距离图，该图中明显拐点位置对应的k距离值设定为Eps。

MinPts 选择

$MinPts$的选择由指导性原则，$MinPts \geq dim + 1$，dim表示数据的维度。如果$MinPts$设为 1，则每个点各自成簇，如果$MinPts$设为 2，则与层次聚类最近邻与结果相同。因此$MinPts \geq 3$。若该值选取过小，则稀疏簇中结果由于密度小于$MinPts$，从而被认为是边界点儿不被用于在类的进一步扩展；若该值过大，则密度较大的两个邻近簇可能被合并为同一簇。因此，该值是否设置适当会对聚类结果造成较大影响。

实验过程

读取数据

定义 loadDataSet()，用于数据集的载入。因为给的数据文件是.mat 文件，使

用 scipy.io.loadmat 导入数据文件，但是要读取里面的数据，需要找到对应的 key。如 2d4c 里，key 有‘a’, ‘moon’, ‘smile’, ‘b’，而 ‘a’ 才是对应需要的数据。

DBSCAN 算法部分

找出 epslion 范围内的点

定义 dist()，计算欧氏距离。
定义 eps_neighbor()，判断两个数据点之间的距离是否在 eps 范围内。
定义 region_query()，遍历数据集里所有点，把在 eps 范围内的点找出。

判断是否进行簇扩张

定义 expand_cluster()，dbscan()，不满足 MinPts 条件的列为噪声点，满足的点划分到该簇，如果是核心点，继续访问他们的邻居进行扩张。遍历所有的点，重复操作，返回聚类簇 id 和聚类簇数目。

eps 选择

定义 select_eps()用于找出最适 eps。对于每个数据点，计算其中某个点与其他数据点的欧氏距离之和，按距离进行排序，找出第 k 个近邻到该数据点的距离，添加到 k_dist 序列中，之后 k-distance 图需要。

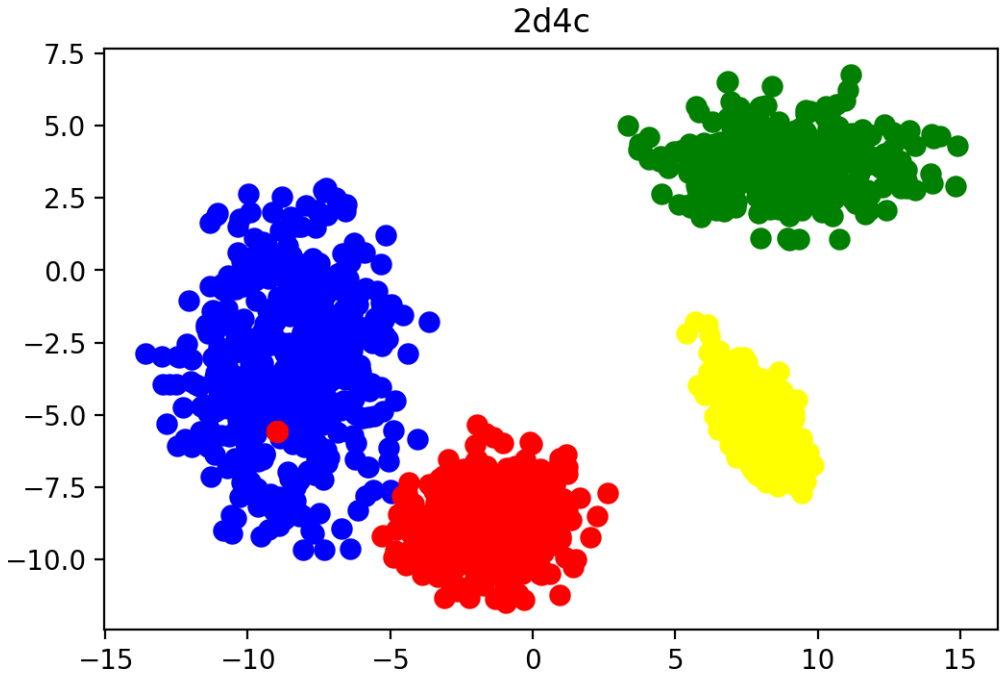
绘图部分

定义 plotFeature(), 用于对就聚类结果的绘图。每个独特的簇 id 用同一种颜色标出。有几个不同的聚类簇，就有多少种颜色显示。

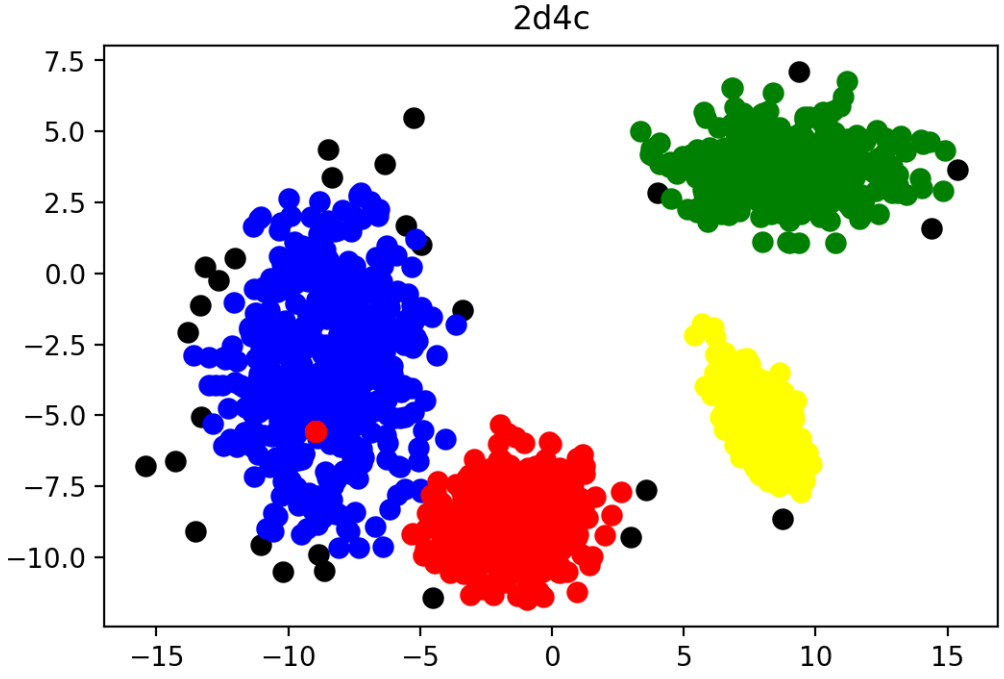
依次对 ‘2d4c’, ‘long’, ‘moon’, ‘sizes5’, ‘smile’, ‘spiral’, ‘square1’, ‘square4’,数据集进行数据读取，DBSCAN 聚类，绘制 k-distance 图，绘制聚类图。

实验结果与分析

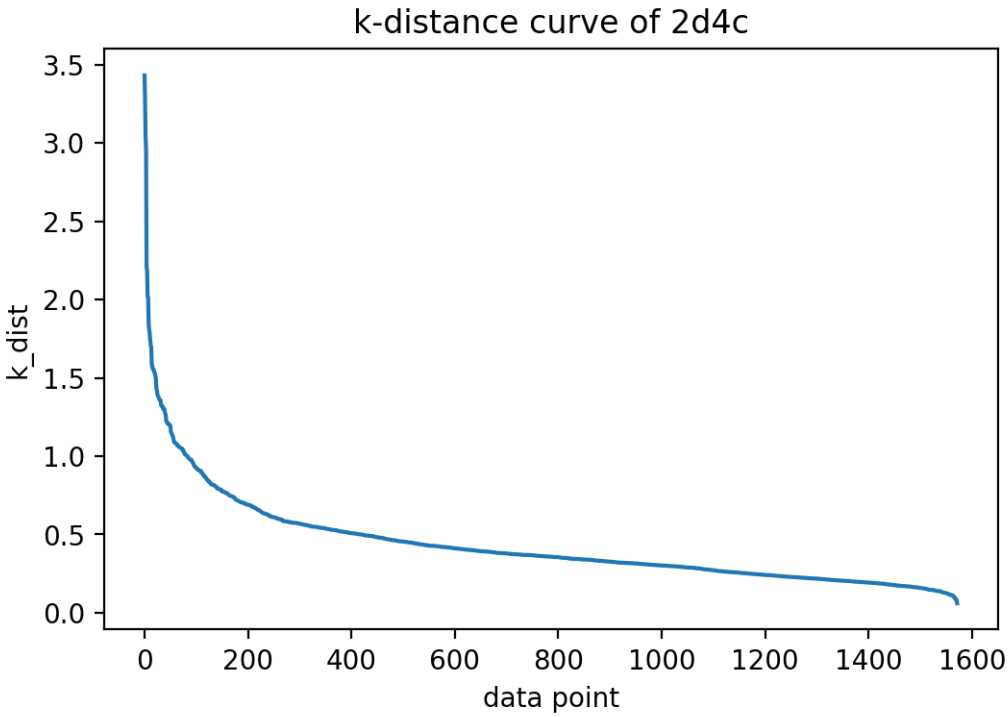
2d4c



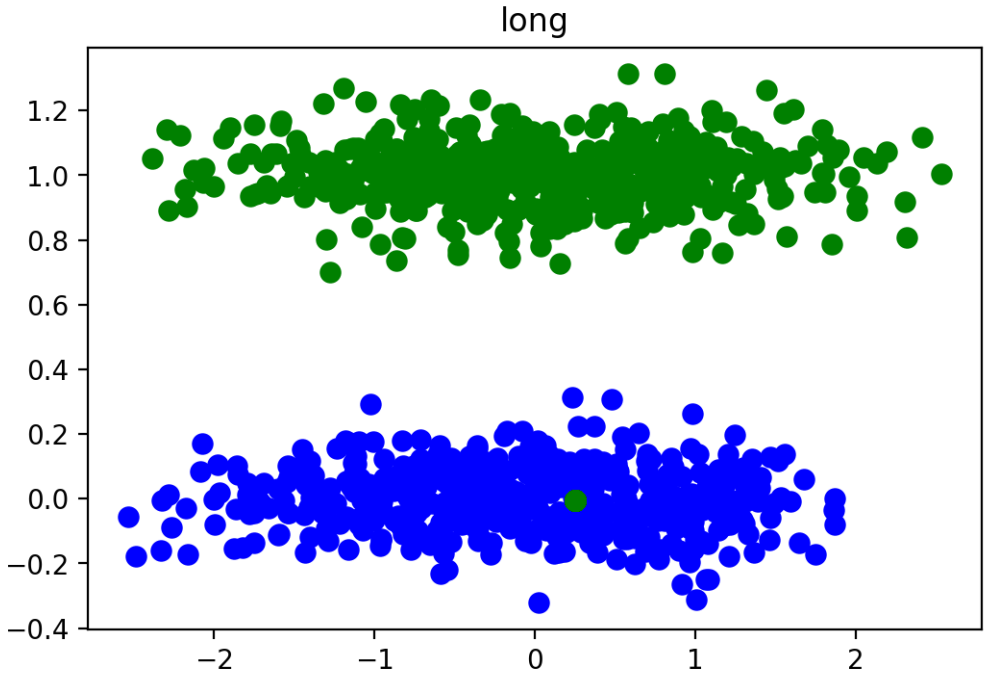
(noise = 0.1)



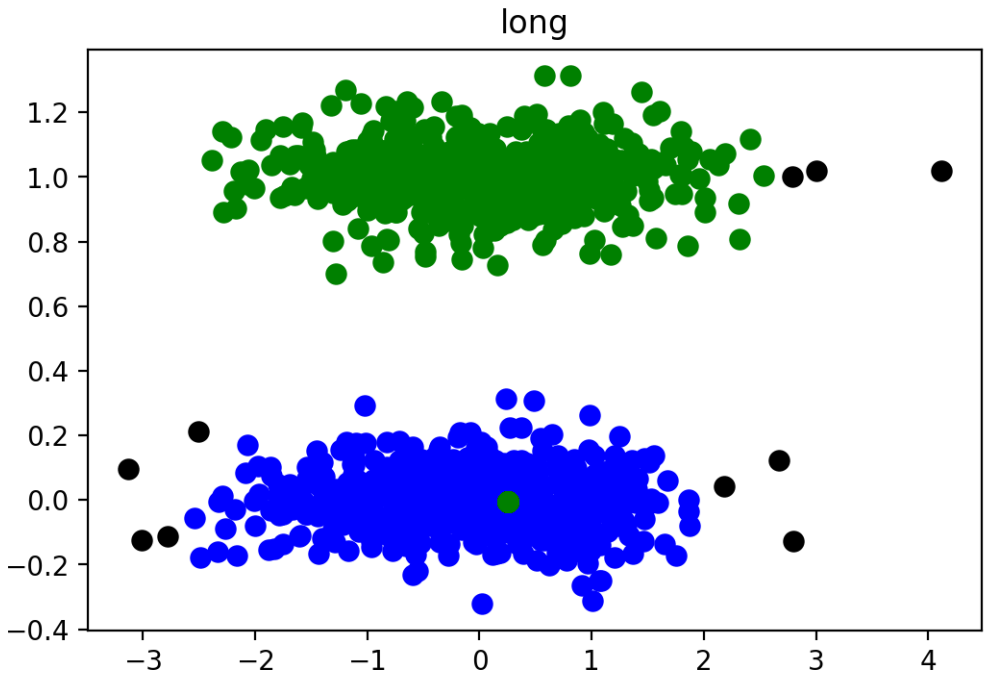
(noise = 0)



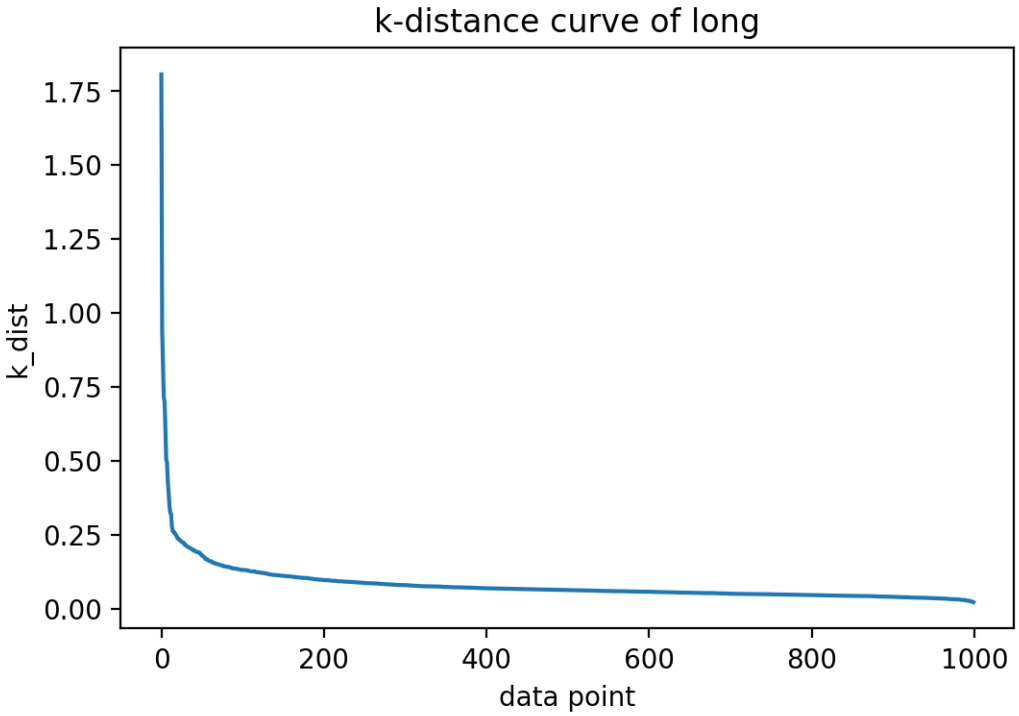
Long



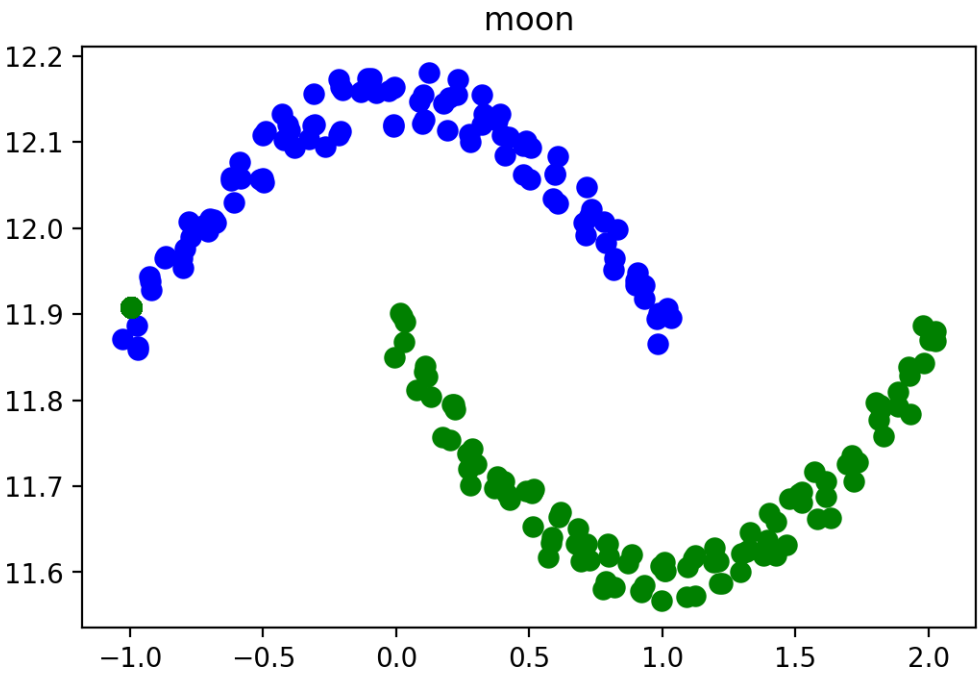
(noise = 0.1)


long
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−3
−2
−1
0
1
2
3
4

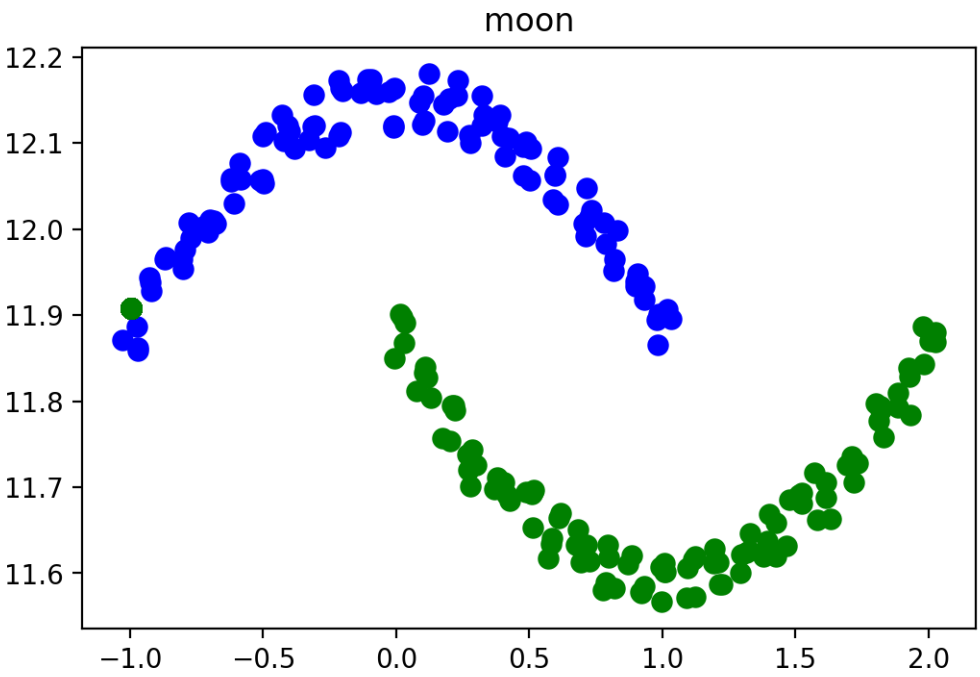
(noise = 0)


k-distance curve of long
k_dist
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000
data point

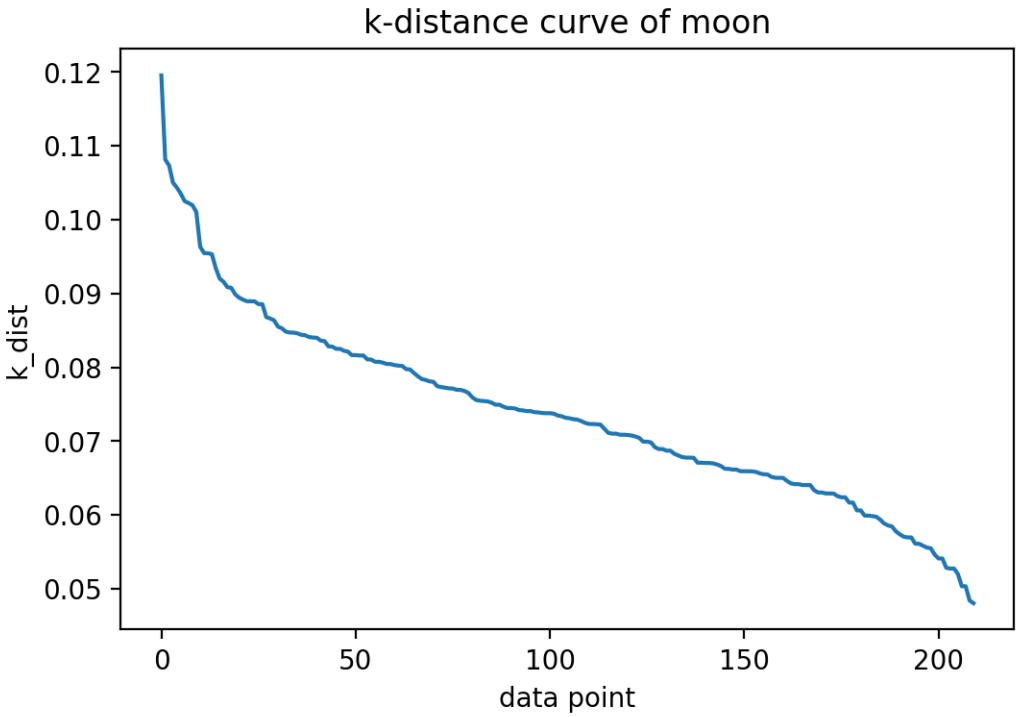
Moon



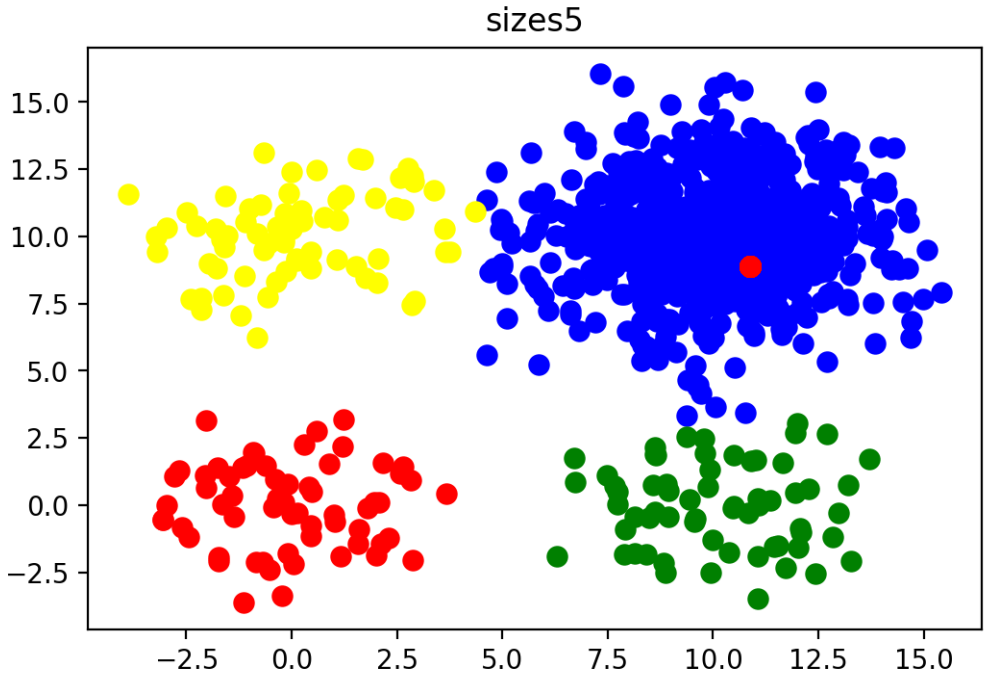
(noise = 0.1)



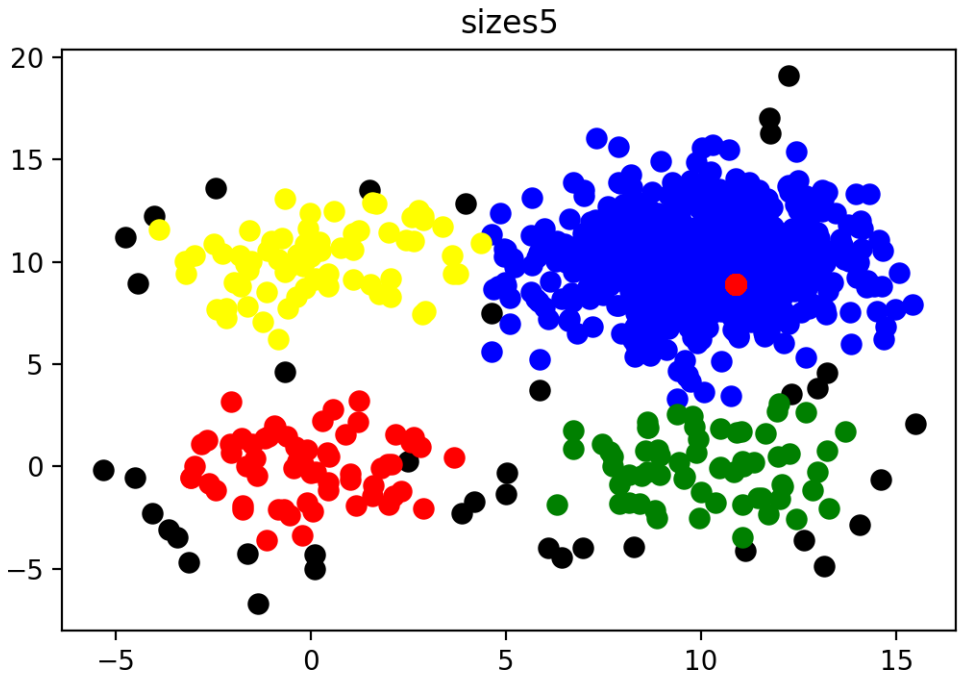
(noise = 0)



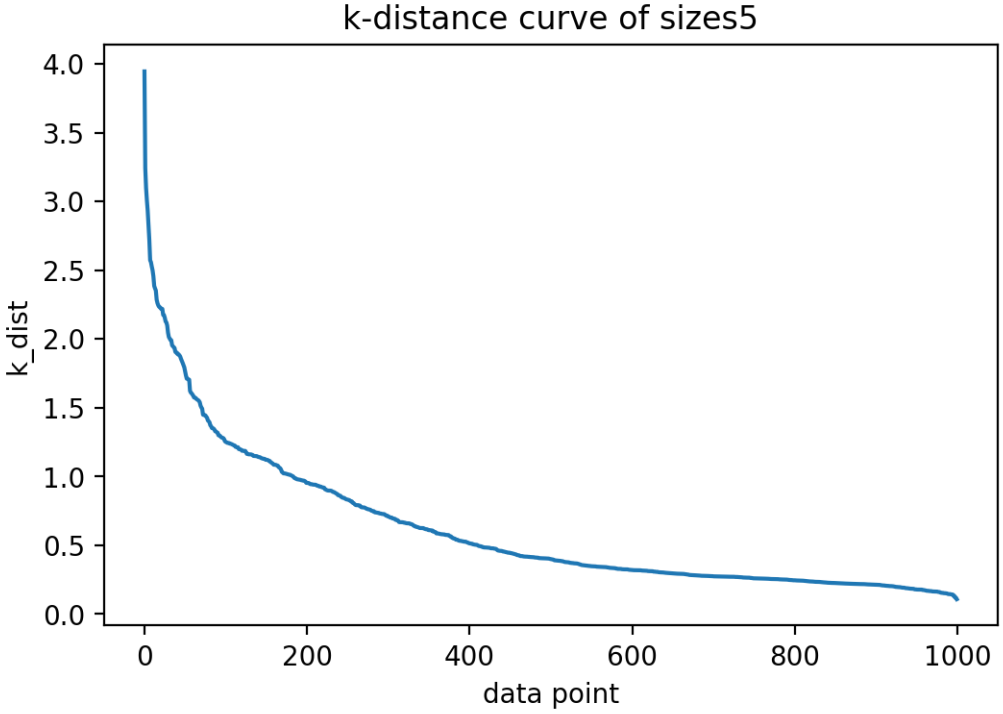
Sizes5



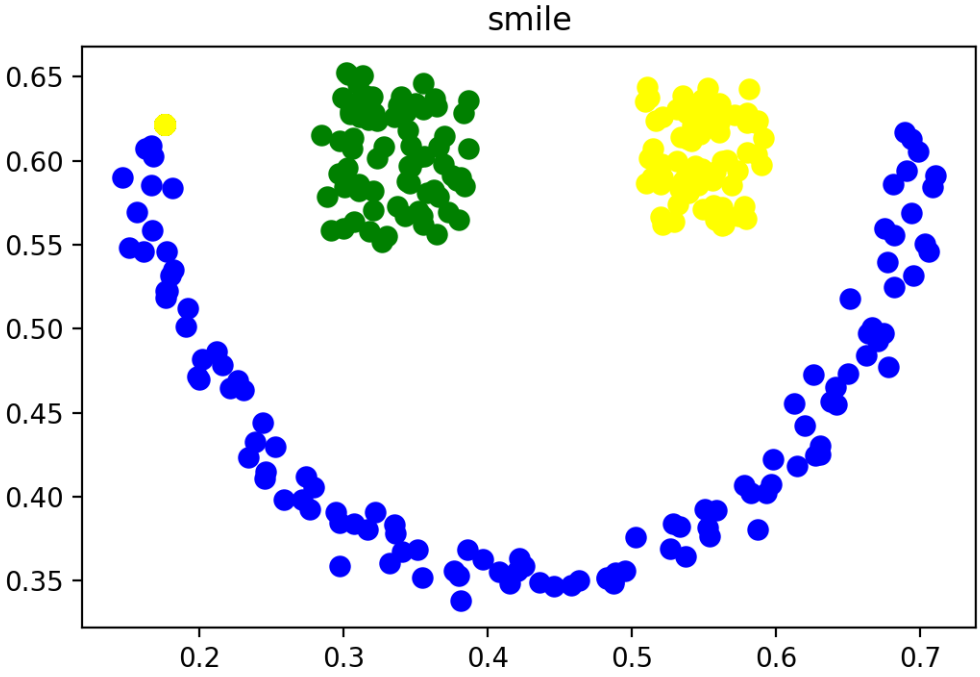
(noise = 0.1)


sizes5
20
15
10
5
0
−5
−5
0
5
10
15

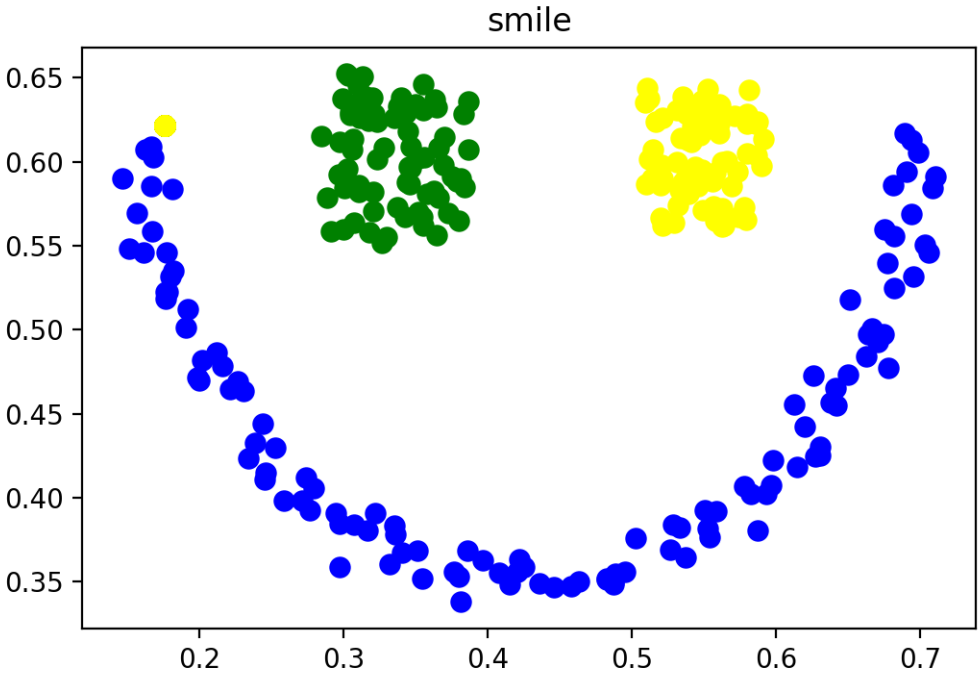
(noise = 0)


k-distance curve of sizes5
k_dist
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
200
400
600
800
1000
data point

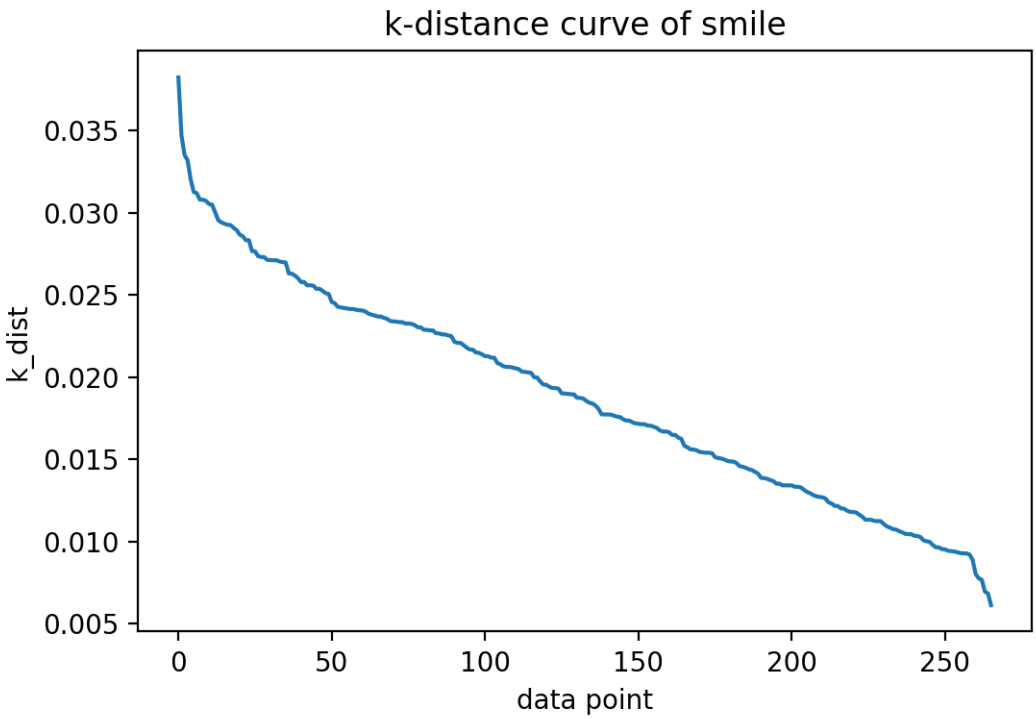
Smile



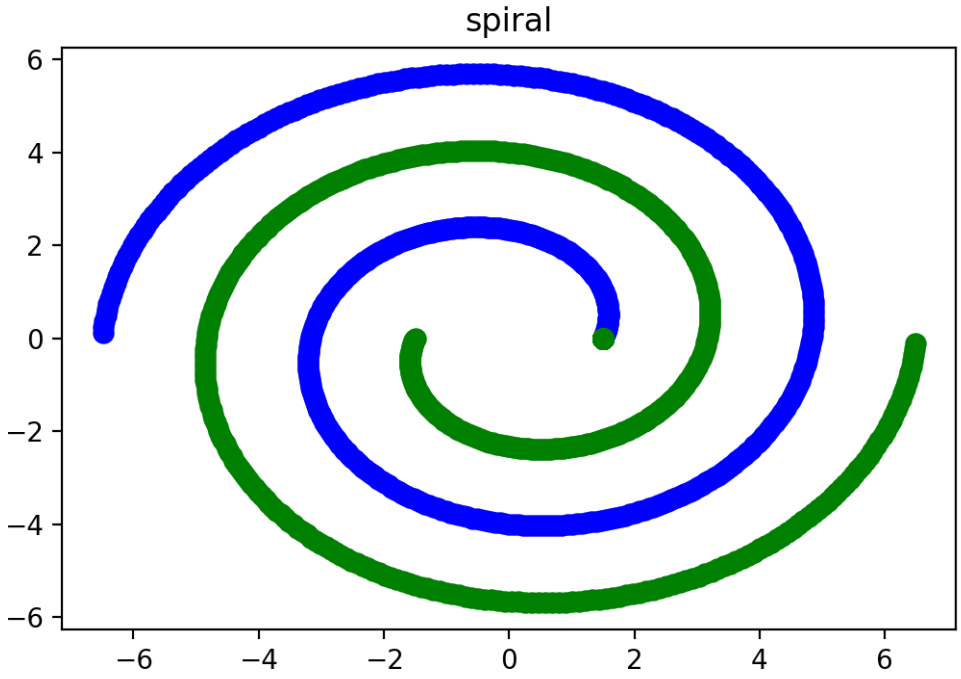
(noise = 0.1)



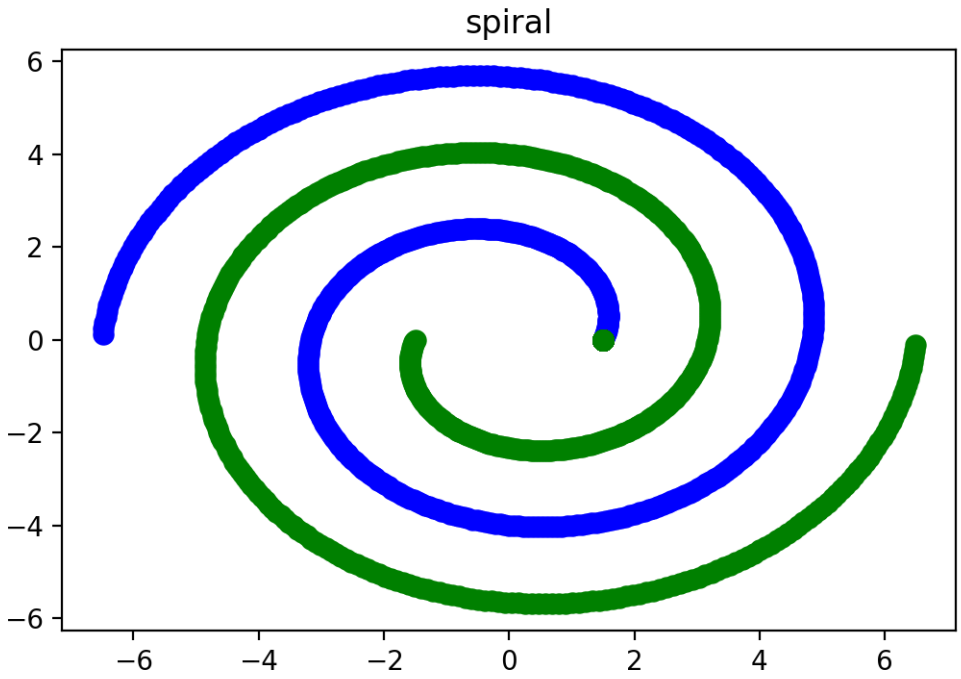
(noise = 0)



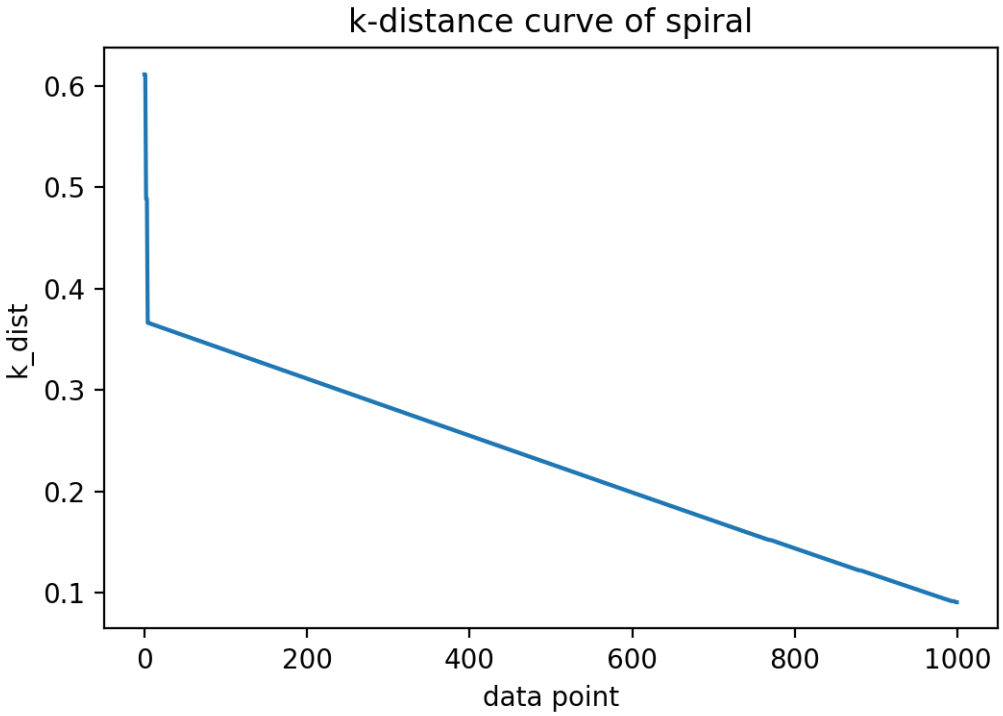
Spiral



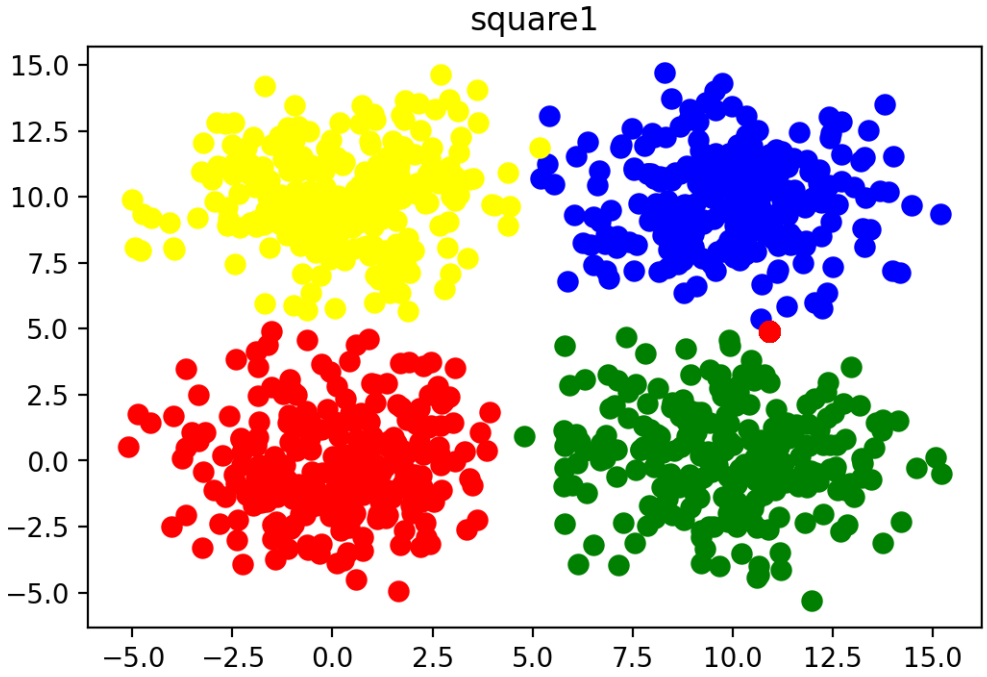
(noise = 0.1)


spiral
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

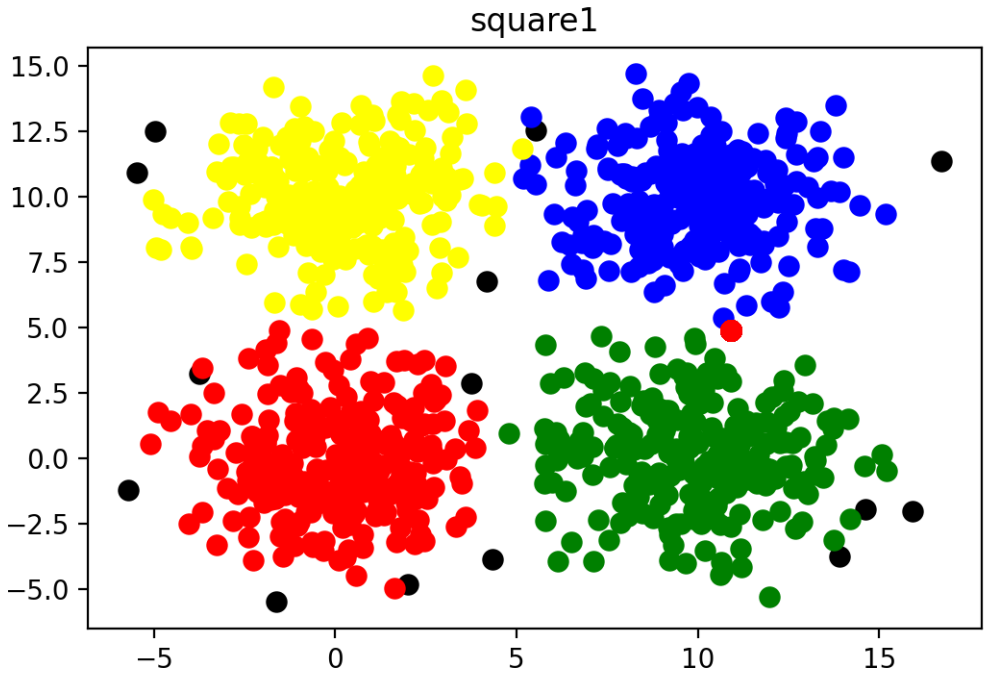
(noise = 0)


k-distance curve of spiral
0.6
0.5
0.4
0.3
0.2
0.1
k_dist
0
200
400
600
800
1000
data point

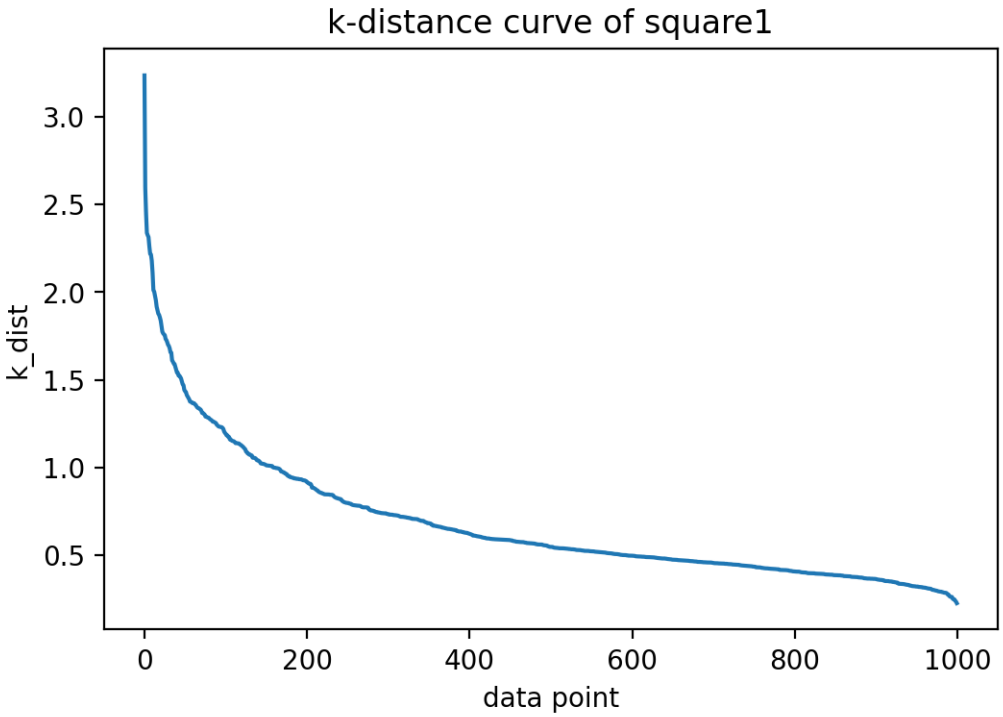
Square1



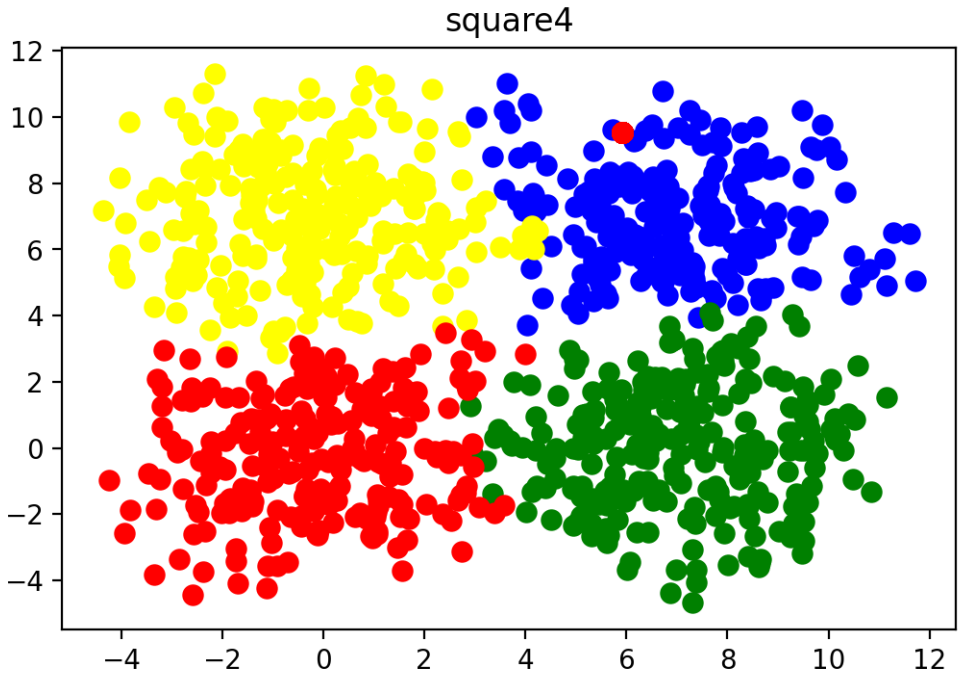
(noise = 0.1)



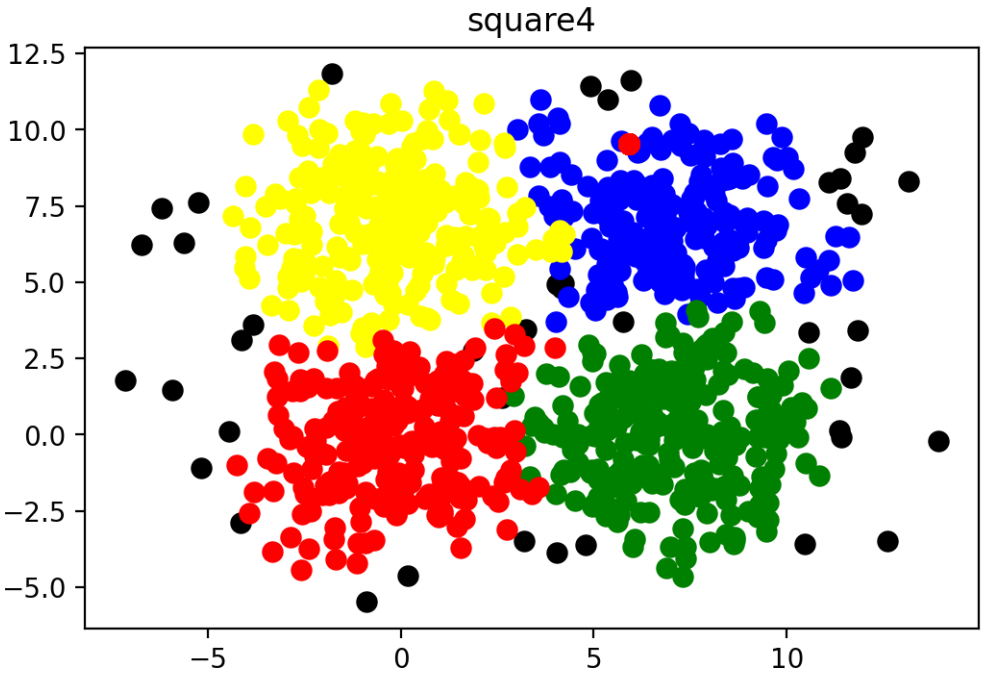
(noise= 0)



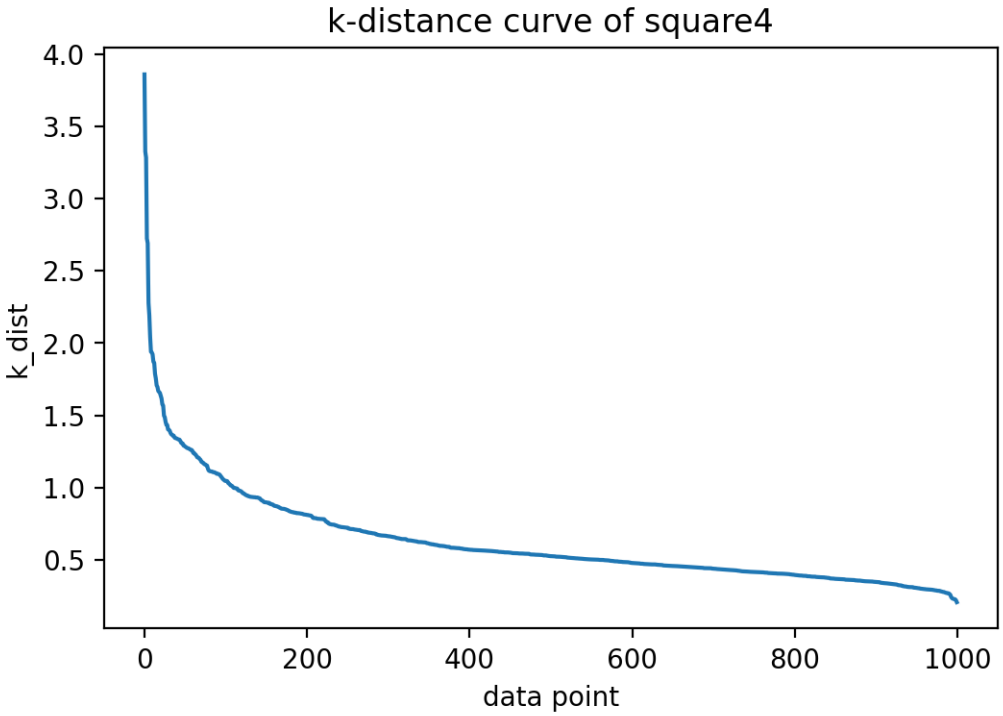
Square4



(noise = 0.1)



(noise = 0)



结果分析

1. 对于‘moon’，‘smile’，‘spiral’这三个分离明显，聚类间差距较大的数据集，不管添不添加噪声，所得到的聚类结果都很好。但是对于‘2d4c’，‘long’，‘sizes5’，‘square1’，‘square4’这五个数据点分布较复杂，密度不均匀的数据集，不添加噪声点的情况下，会遗留有有未归类的噪点，但是对原始数据加了 0.1 的噪声后，反而聚类结果变得更好。

2. 对于 2d4c’，‘sizes5’，‘square1’，‘square4’这四个密度分布不均匀，簇间界限模糊的数据集，eps 与 MinPts 的微调都会对聚类结果造成较大的影响，尤其是未加噪声的原始数据集。
3. 比较奇怪的是，几乎所有的聚类图里，都会有本不属于这一簇的点出现在该簇中，而且是仅有一个点，如月亮的月牙尖角处，笑脸的嘴角处，螺旋曲线最内部，都发现有突兀点，暂时找不到是哪出了问题。

附录

DBSCAN.html