Machine Learning Lab4

Xiaoma

2022 年 11 月 24 日

实验要求

1. 实现 DPC 算法
2. 画出决策图与结果图，计算评价指标 DBI

实验原理

经典的聚类算法 $K-means$ 是指通过指定聚类中心，再通过迭代的方式更新聚类中心的方式，由于每个点都被指派到距离最近的聚类中心，所以导致其不能检测非球面类别的数据分布。虽然有 $DBSCAN$ 对于任意形状分布的进行聚类，但是必须指定一个密度阈值，从而去除低于此密度阈值的噪点。

基于以上分析，DPC 算法基于这样的假设：**聚类中心周围都是密度比其低的点，同时这些点距离该聚类中心的距离比其他聚类中心更近。**

计算局部密度

1. Cut-Off Kernel:

$$\rho_i = \sum_{j \in I_{S/\{i\}}} \chi(d_{ij} - d_c)$$

其中函数

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}$$

参数 d_c 为截断距离，需要事先确定。文章指出，d_c 可以选择为平均每个点的邻居密度为数据总数的 $1\% - 2\%$ 时的阈值距离。

2. Gauss Kernel:

$$\rho_i = \sum_{j \in I_{S/\{i\}}} e^{-(\frac{d_{ij}}{d_c})^2}$$

可知，Cut-Off 为离散值，Gauss Kernel 为连续值，因此相对来说后者产生密度值冲突的概率更小。

计算相对距离

$$\sigma_i = \begin{cases} \min_{j \in I_S^i}\{d_{ij}\}, & I_S^i \neq \emptyset \\ \max_{j \in I_S}\{d_{ij}\}, & I_S^i = \emptyset \end{cases}$$

- 对于非局部密度最大点：
  - 找到所有局部密度比 i 点高的点
  - 这些点中与 i 距离最近的点的距离就是 σ_i
- 对于局部密度最大点：
  与该点距离最大点的距离就是 σ_i

寻找聚类中心

根据计算得到的 σ, ρ，绘制决策图，将图中的点进行分割，选择拥有较高的 σ, ρ 的点作为聚类中心，而拥有较高 σ，较低 ρ 的点被视为异常点。

剩余点的类别分派

当前点的类别与比它局部密度大且与其相对距离最近的点的类别相同

去除噪点

首先为每个类定义一个边界区域，即分配到该类别但与其他类别的点距离小于 d_c 的点的集合，找到每个类别边界区域中密度最高的点，以该点的密度作为阈值来筛选类别。

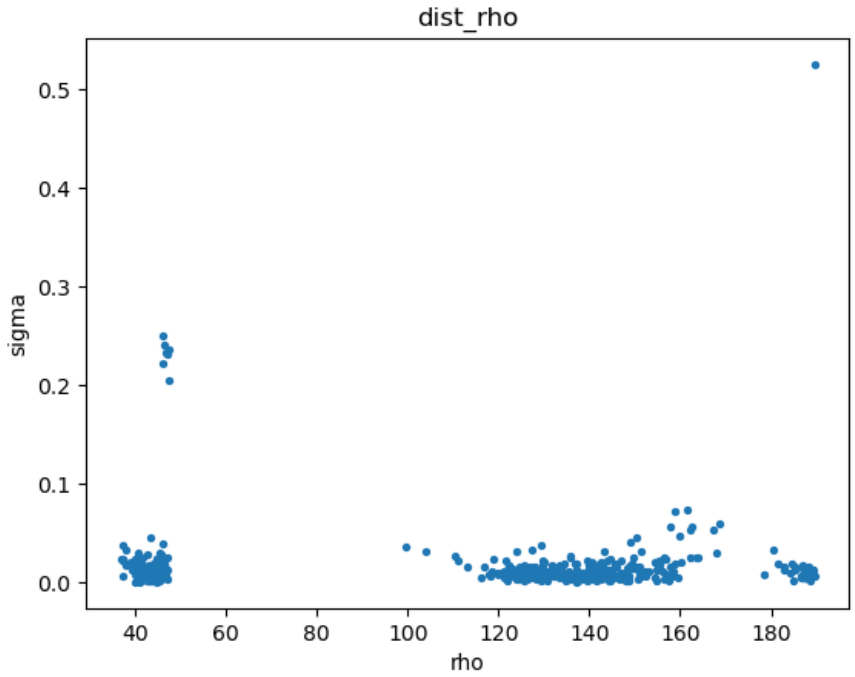
实验步骤

截断距离的选择

自动计算截断距离

根据论文所描述的，将截断距离选取为满足使所有点在截断距离内都包含 $1\% - 2\%$ 的情况，得到结果为

R15 数据集，d_c = 0.19

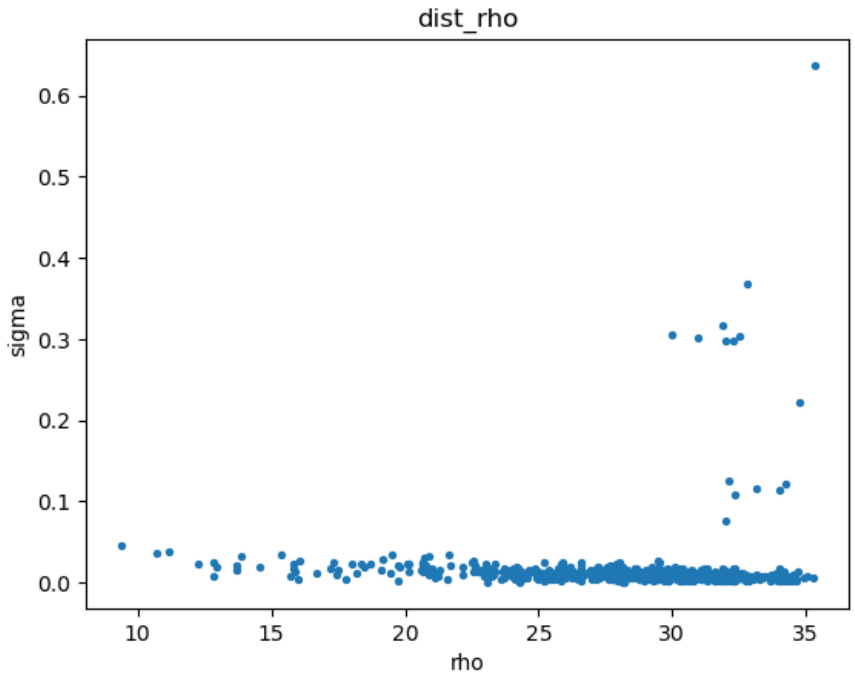


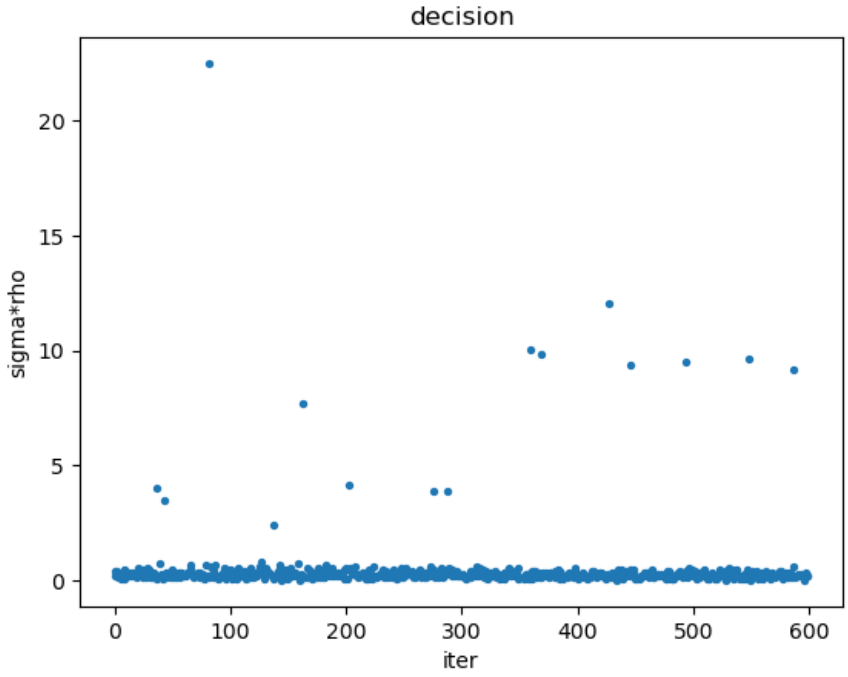
无法对决策图中的点进行合理分割，故采用手动设置截断距离的方法

手动设置阶段距离

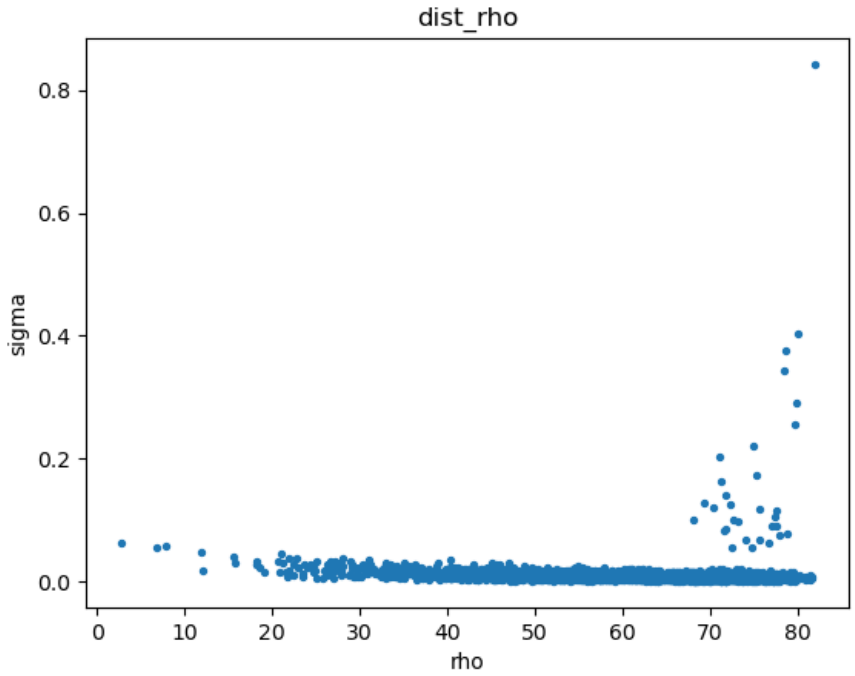
通过多次调试，大致确定使相应数据集决策图表现较好的截断距离

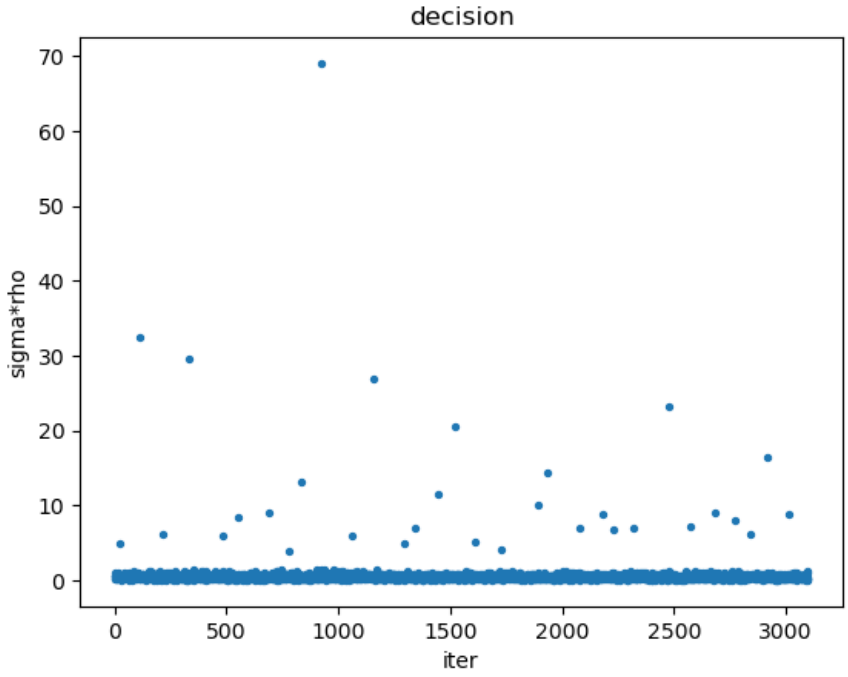
R15 d_c = 0.06



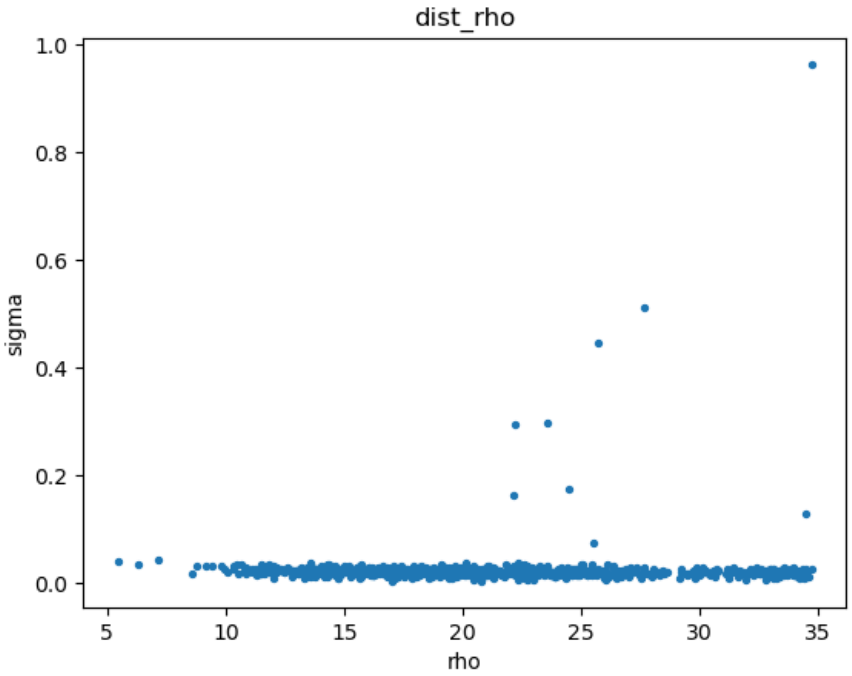

decision
sigma*rho
iter
0
5
10
15
20
0
100
200
300
400
500
600

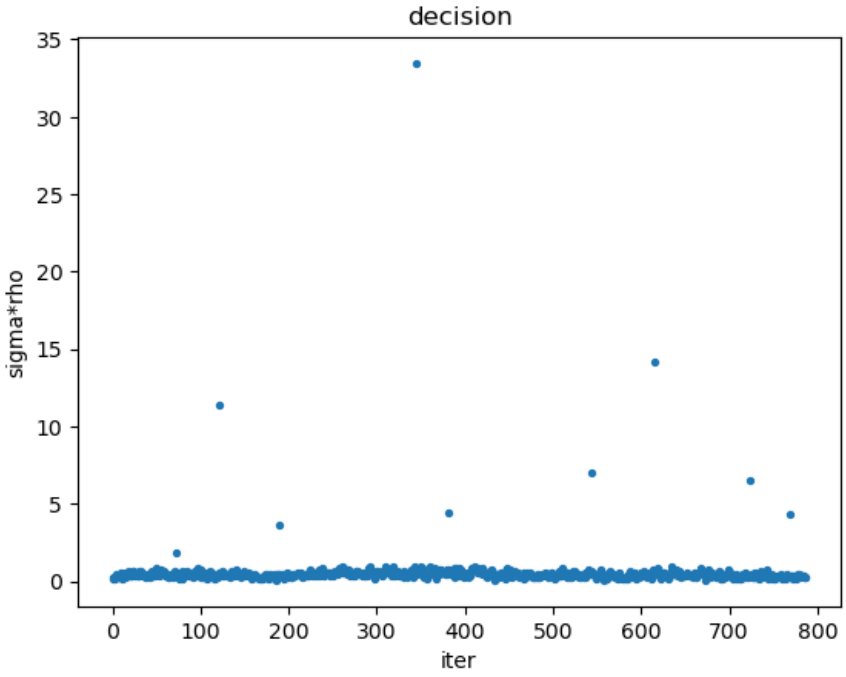
D31 d_c = 0.06


dist_rho
sigma
rho
0.0
0.2
0.4
0.6
0.8
0
10
20
30
40
50
60
70
80



Aggregation d_c = 0.075





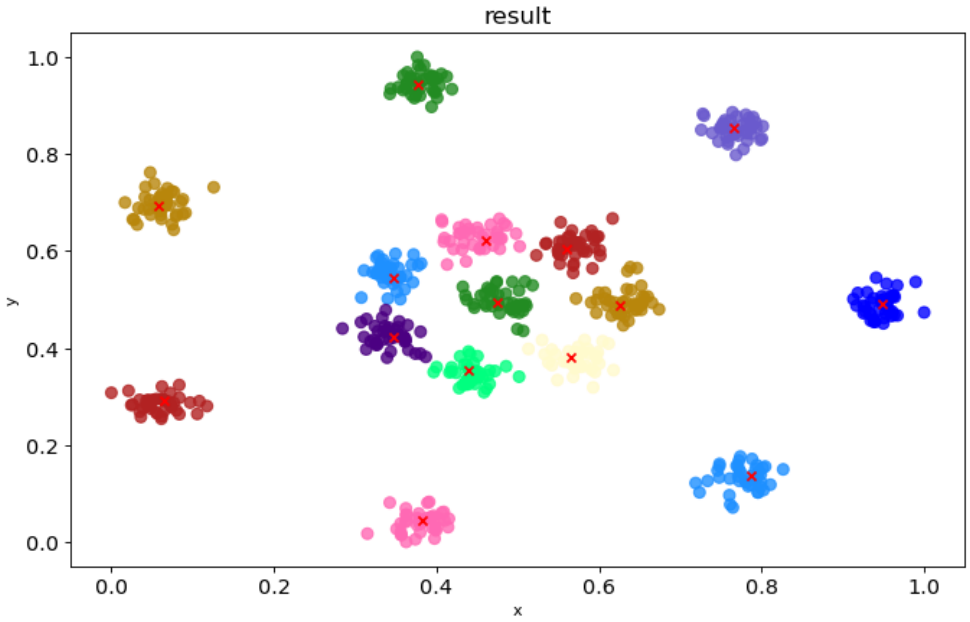
将 $sigma-rho$ 图与 $simga*rho-iter$ 图进行相互验证，效果较好。

对数据集进行分类 (若有异常点或噪点，颜色为黑色)

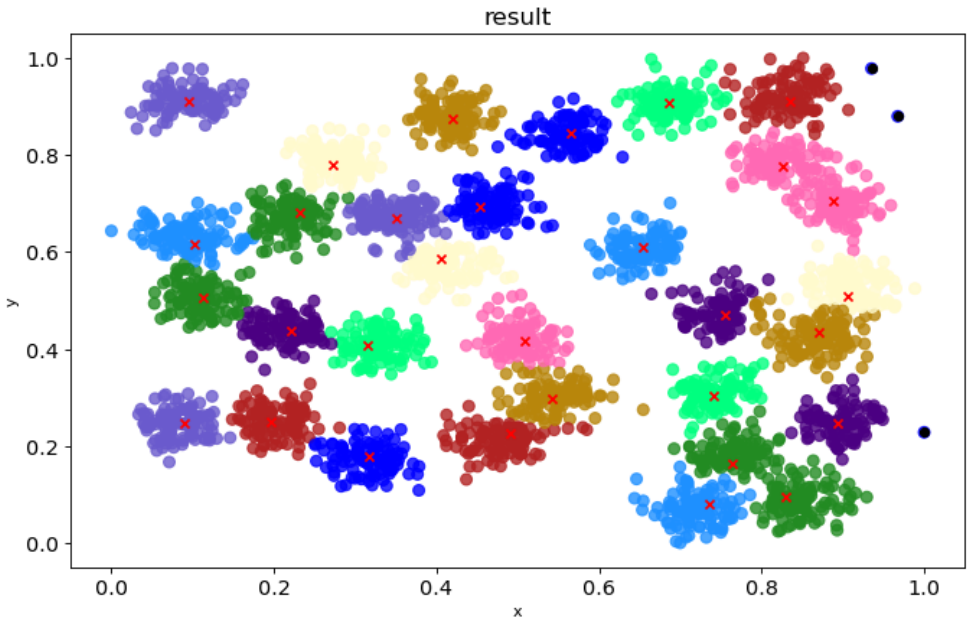
根据决策图确定密度阈值与相对距离阈值

R15 center_rho = 29.5 center_sigma = 0.07

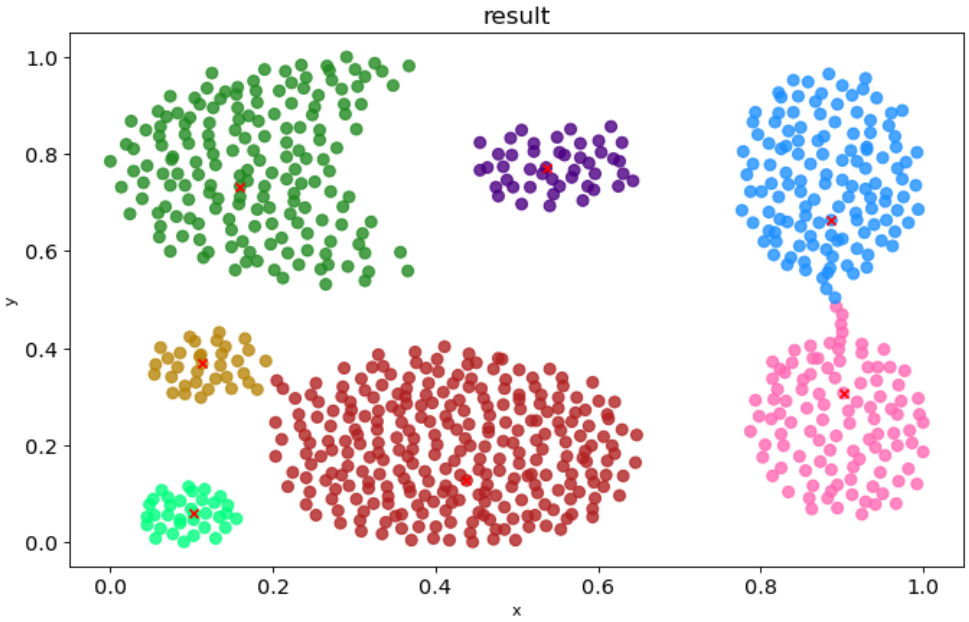
DBI = 0.31471445116423397



D31 center_rho = 67.9 center_sigma = 0.05
DBI = 1.0364455020740102



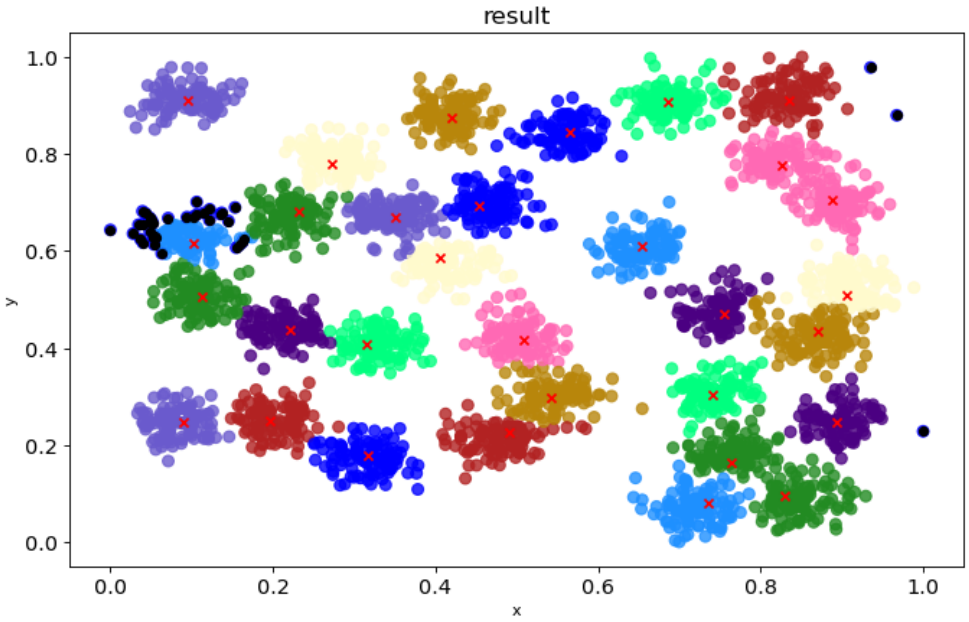
Aggregation center_rho = 21.9 center_sigma = 0.155
DBI = 0.5435124431628843



是否使用去除噪点

通过测试不同的 d_c，发现去除噪点并不总是能产生更好的结果，d_c 值过大会使超出边界的点过多，过小会导致划分不出超出边界的点，经过比较使用不同 d_c 与比较是否去除噪点发现，只有 D31 数据集使用了去除噪点后表现更好

D31 center_rho = 67.9 center_sigma = 0.05 odd_d_c = 0.01
DBI = 0.828635511619769（未去噪点时为 1.0364455020740102)



当 kernel='cut off' 时与高斯核实验结果基本相同，故在实验报告中省略。

实验分析

与其他机器学习库的比较

DBI	myDPC	kmeans	DBSCAN
R15	0.315	0.315	5.232
D31	0.829	0.722	1.353
Aggregation	0.544	0.776	0.625

根据比较发现，kmeans 对三种数据集的分类结果都比较好，而 DBSCAN 值对 Aggregation 数据集分类效果较好，可以大致认为，DPC 算法集成了两种算法的优点，但不足之处在于 d_c 的选取需要手动确定