DBSCAN

【概述】

DBSCAN是Density-Based Spatial Clustering of Applications with Noise的缩写，是一种简单有效的基于密度的聚类算法。

基于密度的聚类旨在检测高密度的区域，这些区域由低密度的区域相互分开。

【算法原理】

应用DBSCAN算法时，我们需要估计数据集中特定点的密度，特定点的密度是通过计算该点在指定半径下数据点的个数（包括特定点），这种计算得到的某个点的密度也被称为局部密度。

通过上述方式计算的点的密度取决于指定的半径：假设数据集点的个数为m，如果半径足够大的话，那么所有点的密度都是m；如果半径太小的话，所有点的密度都是1（即为本身这个点）。

计算数据集中每个点的密度时，我们需要把每个点归为一下三类：

1、点在一个密集区域的内部：归为核心点。

2、在密集区域的边缘：归为边界点。

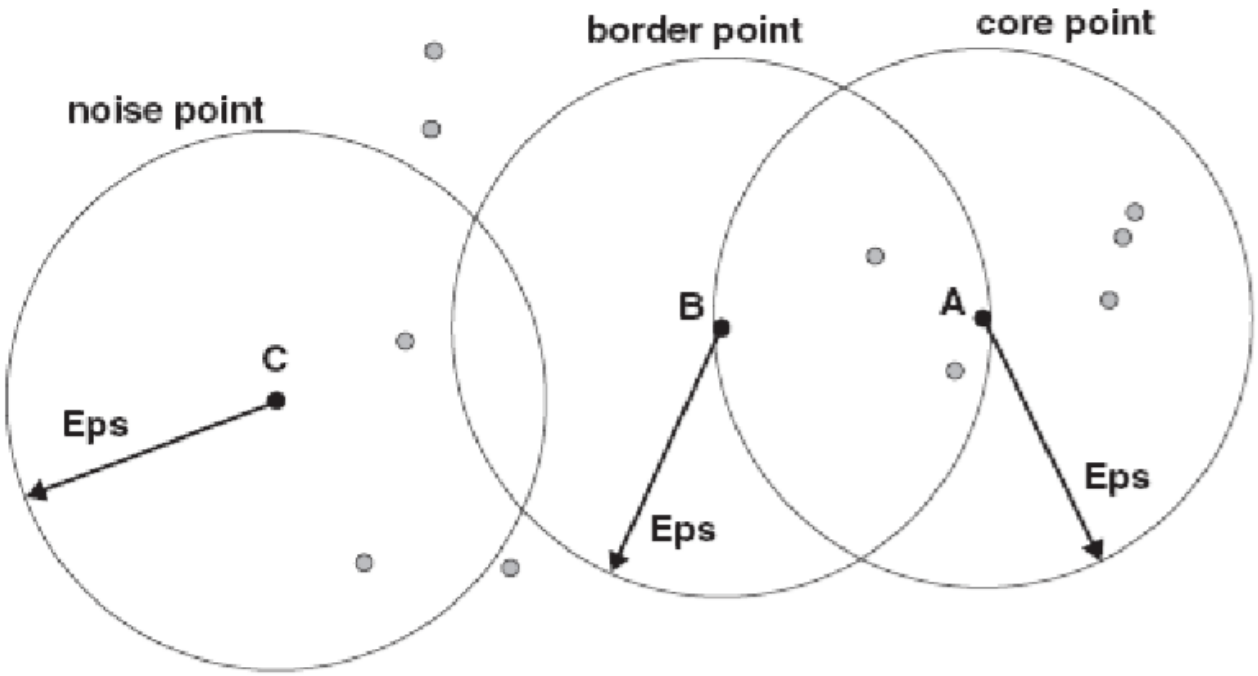
3、在以一个稀疏地区：归为噪声点。

下面是核心点、边界点以及噪声点的定义：

核心点：如果该点的局部密度大于某个阈值，称这个点为核心点。下图中，A是一个核心点。

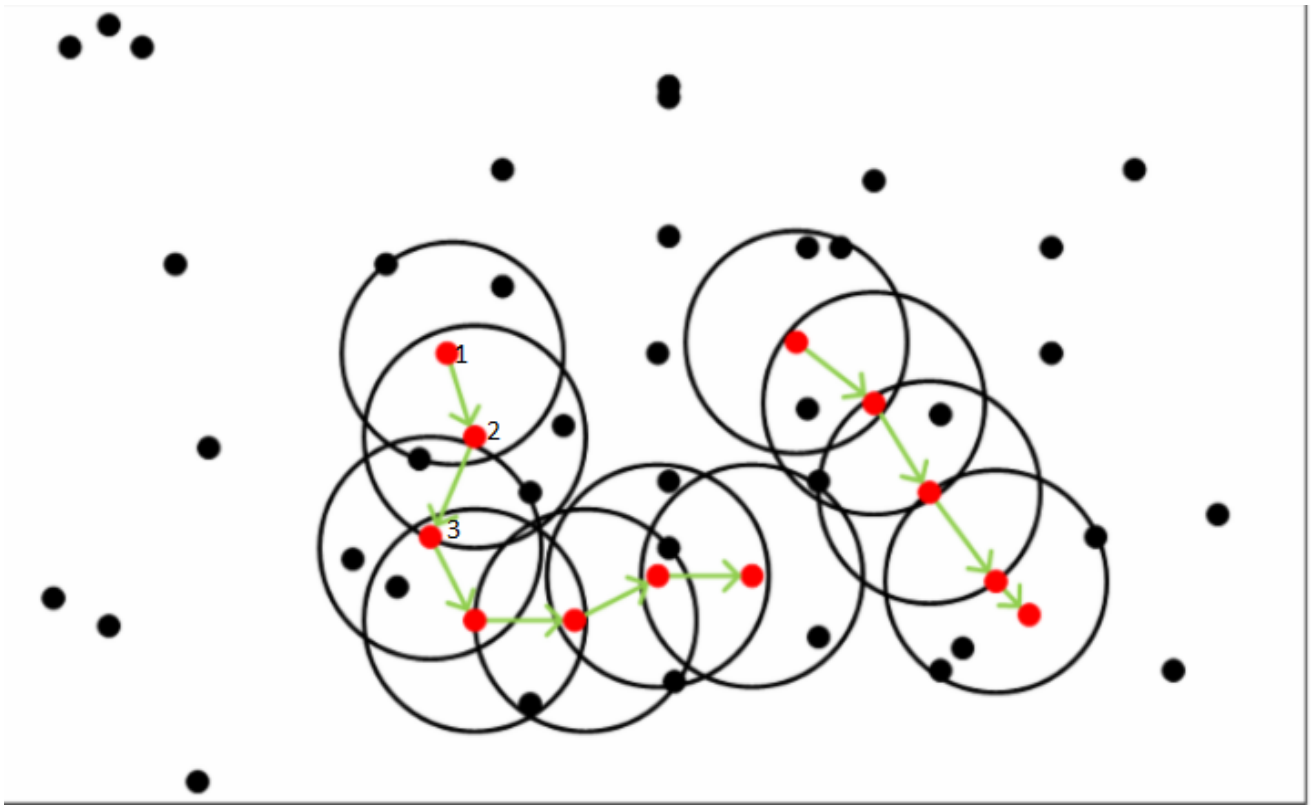
边界点：该点的局部密度小于某个阈值，但是它落在核心点的领域内。邻域通过半径和距离公式定义，在下图中即为以核心点为圆心的圆内。下图中，B是一个边界点。

噪声点：不属于核心点也不属于边界点的点。下图中，C是一个噪声点。



除了标记数据集中每个点的类别，我们要做的是每个点的归属类别。对于同一个还未分配的核心点，我们将它邻域内的所有点归为一个新的类C_{new}。如果邻域内有其他核心点的话，我们需要将其他核心点邻域内的还未分配的点也分配给这个新类，如果其他核心点邻域内也包含其他新的核心点，我们将重复上面相同情况的动作。

举个例子，下图核心点1邻域内的点归为类C_{new}，因为邻域包括核心点2，那么核心点2邻域内的未分配类别的也归类为C_{new}，核心点2邻域包含核心点3，核心点3也进行与核心点2相同的操作，依次类推（图中绿箭头标识的过程）直到核心点邻域内不包含新的核心点。



【算法流程】

- 如果所有点已经处理，停止。
- 对于以前没有处理的特定点，检查它是否是核心点。
- 如果不是核心点
    - 将其标记为噪声点（此标签可能稍后会更改）
- 如果是核心点，将其标记并
    - 使用这一点形成一个新的聚类C_{new}，并包括集群内的Eps-邻域内或边界上的所有点。
    - 将所有这些在邻域内的点插入队列中。
    - 当队列不为空时
        - 从队列中删除第一个点
        - 如果这一点不是一个核心点，将其标记为边界点。
        - 如果这个点是核心点，则标记它并检查其邻居中以前没有分配给类的每个点。对于每一个未分配的相邻点
            - 将该点分配给当前类C_{new}
            - 将该点插入队列