对大数据集的基于密度的算法——DBSCAN

1、DBSCAN 算法概述

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的聚类算法。与划分和层次聚类方法不同，它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。

2、DBSCAN 算法原理

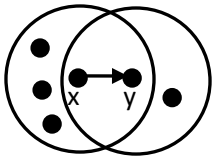
DBSCAN 中的几个**概念**：

ε：以点 x 为中心，ε 为半径，组成的超球体区域为 $V_\varepsilon(x)$。

q：区域 $V_\varepsilon(x)$ 中的点数最少为 q。

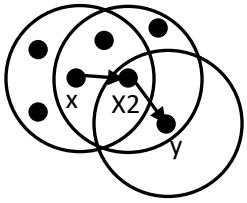
核心点：如果以点 x 为中心，区域 $V_\varepsilon(x)$ 中的点数大于等于 q，则称点 x 为核心点。

直接密度可达：如果 $y \in V_\varepsilon(x)$，并且点 x 是核心点，则称点 x 到点 y 是直接密度可达的。



上图中，$q=5$。点 x 到点 y 直接密度可达，但点 y 到点 x 不是直接密度可达的。

密度可达：如果存在顺序点 $x_1, x_2, ..., x_p$，满足 $x_1 = x$，$x_p = y$，并且点 x_i 到点 x_{i+1} 是直接密度可达的，则称点 x 到点 y 是密度可达的。



上图中，$q=5$。点 x 到点 x2 直接密度可达，点 x2 到点 x 直接密度可达，点 x2 到点 y 直接密度可达。点 x 到点 y 密度可达，但点 y 到点 x 不是密度可达的。

密度连通：如果存在点 z，点 z 到点 x 和点 y 是密度可达的，那么点 x 和点 y

是密度连通的。

DBSCAN 算法：（X_{un} 是没有被聚类的点，m 为聚类个数）

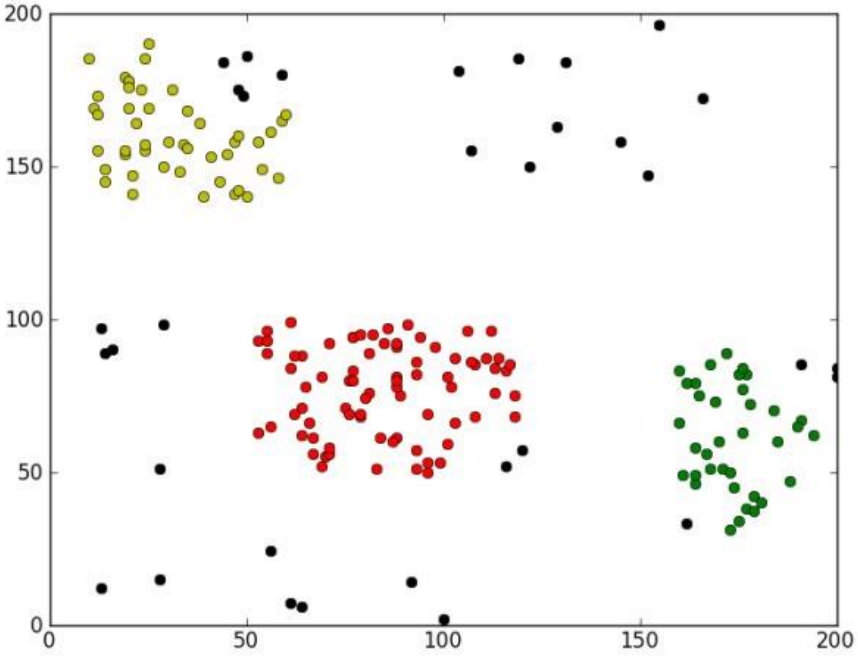
- 令 $X_{un} = X$
- 令 $m = 0$
- While $X_{un} \neq \phi$ do

（1）任意选择一个 $x \in X_{un}$

（2）如果 x 是一个非核心点，则把 x 标记为噪声点，$X_{un} = X_{un} - \{x\}$

（3）如果 x 是一个核心点，那么：$m = m + 1$，确定 X 中所有与 x 密度可达到的点，并把找到的这些点和 x 分配到 C_m 聚类（可能之前标记为噪声的边界点也划分到 C_m 聚类中）。$X_{un} = X_{un} - C_m$。

下图是一个用 DBSCAN 算法进行聚类的例子，其中黑色的为噪声点。



3、DBSCAN 算法优缺点

优点：

（1）DBSCAN 算法不需要像 k-means 算法一样，要指定簇的个数。

（2）DBSCAN 可以聚出任意形状的簇，它甚至可以找到被另一个簇包围（但

不连起来的）的簇，因为有 MinPts 参数，single-link 现象（不同的簇被一个点连起的细线连起来了）被降低了。

（3）DBSCAN 有噪音的概念，可以有效的处理异常数据。

（4）DBSCAN 只需要两个参数，并且对数据库中点的排序顺序不敏感。

（5）算法的时间复杂度低于$O(N^2)$，更适合处理大数据集。

缺点：

（1）不适用高维数据集，DBSCAN 的质量取决于 regionQuery(P,Eps)函 数中距离的测量。最常用的距离度量是欧式距离，尤其是在高维数据中，由于所谓的维数灾难，这种度量基本上是无用的，很难为 E 找到一个恰当的值

（2）DBSCAN 算法将区域查询得到的所有未被处理过的点都作为种子点，留待下一步扩展处理。对于大规模数据集中的较大类而言，这种策略会使种子点的数目不断膨胀，算法所需的内存空间也会快速增加。）

改进：

通过选用核心点邻域中的部分点作为种子点来扩展类，从而大大减少区域查询的次数，降低 I/O 开销，实现快速聚类。DBSCAN 的改进算法 OPTICS 克服了必须仔细选择参数q和ε的缺点。

（3）由于 DBSCAN 算法只设置了一组参数，因此当各个类的密度不均匀，或类间的距离相差很大时，聚类的质量较差。(当各个类的密度不均匀、或类间的距离相差很大时，如果根据密度较高的类选取较小的 Eps 值，那么密度相对较低的类中的对象 Eps 邻域中的点数将小 Minpts，则这些点将会被错当成边界点，从而不被用于所在类的进一步扩展，因此导致密度较低的类被划分成多个性质相似的类。与此相反，如果根据密度较低的类来选取较大的 Eps 值，则会导致离得较近而密度较大的类被合并，而它们之间的差异被忽略。所以在上述情况下，很难选取一个合适的全局 Eps 值来获得比较准确的聚类结果。)

改进：

为了解决上述问题，周水庚等人提出了 PDBSCAN (Partitioning-based DBSCAN) 算法。该算法基于数据分区技术来扩展 DBSCAN 算法，它根据数据的分布特性，将整个数据空间划分为多个较小的分区，然后分别对这些局部分区进行

聚类，最后将各个局部的聚类结果进行合并。 PDBSCAN 的算法思想是：首先，根据数据集在某一维或多个维上的分布特性，将整个数据空间划分为若干个局部区域，使得各局部区域内的数据尽可能分布均匀；然后依次绘制各个局部区域的 k-dist 图，并依次得到各个区域的 Eps 值，接着用 DBSCAN 算法对各个局部区域进行局部聚类；最后，将各个局部聚类的结果进行合并，从而完成整个数据集的聚类分析。由于每个局部区域都使用各自的局部 Eps 值来进行聚类，因此有效缓解了因使用全局 Eps 值而导致的聚类质量恶化的问题。

（4）DBSCAN 不是完全确定的，边界点从不同的簇中获得，可以使不同簇的一部分，取决于数据处理

（5）输入参数敏感,确定参数 Eps , MinPts 困难 ,若选取不当 ,将造成聚类质量下降。尽管 DBSCAN 算法提供了利用绘制降序 k-距离图的可视化方法来选择 Eps ,选定的 Eps 值已经比较接近“理想”值 ; 但常有微小差距 , 最终造成聚类结果的相差很大。可以考虑采用如下方法来加以改善 :

DBSCAN 的扩展叫 OPTICS（Ordering Points To Identify Clustering Structure）通过优先对高密度（high density）进行搜索，然后根据高密度的特点设置参数，OPTICS 并不显示的产生结果类簇，而是为聚类分析生成一个增广的簇排序（比如，以可达距离为纵轴，样本点输出次序为横轴的坐标图），这个排序代表了各样本点基于密度 的聚类结构。从这个排序中可以得到基于任何参数 E 和 minPts 的 DBSCAN 算法的聚类结果。

Optics 聚类的理解

了解 optics 算法之间我们只需要了解两个概念

核心距离：

对象 p 的核心距离是指 p 成为核心对象的最小半径 r,如果 p 不是核心对象，那么 p 的核心距离没有任何意义。

可达距离：

对象 q 到 p 的可达距离是指 p 的核心距离和 p 与 q 之间欧氏距离的最大值，如果 p 不是核心对象，p 和 q 之间的可达距离没有意义。

输入：样本集 D, 邻域半径 E, 给定点在 E 领域内成为核心对象的最小领域点

数 MinPts

输出：具有可达距离信息的样本点输出排序方法：1 创建两个队列，有序队列和结果队列。（有序队列用来存储核心对象及其该核心对象的直接可达对象，并按可达距离升序排列；结果队列用来存储样本点的输出次序）；

2 如果所有样本集 D 中所有点都处理完毕，则算法结束。否则，选择一个未处理（即不在结果队列中）且为核心对象的样本点，找到其所有直接密度可达样本点，如果该样本点不存在于结果队列中，则将其放入有序队列中，并按可达距离排序；

3 如果有序队列为空，则跳至步骤 2，否则，从有序队列中取出第一个样本点（即可达距离最小的样本点）进行拓展，并将取出的样本点保存至结果队列中，如果它不存在结果队列当中的话。

3.1 判断该拓展点是否是核心对象，如果不是，回到步骤 3，否则找到该拓展点所有的直接密度可达点；

3.2 判断该直接密度可达样本点是否已经存在结果队列，是则不处理，否则下一步；

3.3 如果有序队列中已经存在该直接密度可达点，如果此时新的可达距离小于旧的可达距离，则用新可达距离取代旧可达距离，有序队列重新排序；

3.4 如果有序队列中不存在该直接密度可达样本点，则插入该点，并对有序队列重新排序；

5、DBSCAN 算法应用

对于大数据集，DBSCAN 算法具有很快的速度，并且可以有效的去除噪声点。并且，该算法适用于聚类个数未知的情况。不适用聚类密度差异太大的数据。

应用的领域：卫星图像分类、X 射线结晶学原子分类、温度数据异常检测。