西安电子科技大学

智能数据挖掘

题　　目：　DBSCAN 密度聚类

班　　级：　1502051

姓　　名：　鲍超俊

学　　号：　15020510059

指导教师：　缑水萍

一、 DBSCAN 密度聚类算法

1. 算法简介

DBSCAN，英文全写为 Density-based spatial clustering of applications with noise ，是在 1996 年由 Martin Ester, Hans-Peter Kriegel, Jörg Sander 及 Xiaowei Xu 提出的聚类分析算法， 这个算法是以密度为本的：给定某空间里的一个点集合，这算法能把附近的点分成一组（有很多相邻点的点），并标记出位于低密度区域的局外点（最接近它的点也十分远），DBSCAN 是其中一个最常用的聚类分析算法，也是其中一个科学文章中最常引用的。

在 2014 年，这个算法在领头数据挖掘会议 KDD 上获颁发了 Test of Time award，该奖项是颁发给一些于理论及实际层面均获得持续性的关注的算法。

2. 基础知识

考虑在某空间里将被聚类的点集合，为了进行 DBSCAN 聚类，所有的点被分为核心点，(密度)可达点及局外点，详请如下：

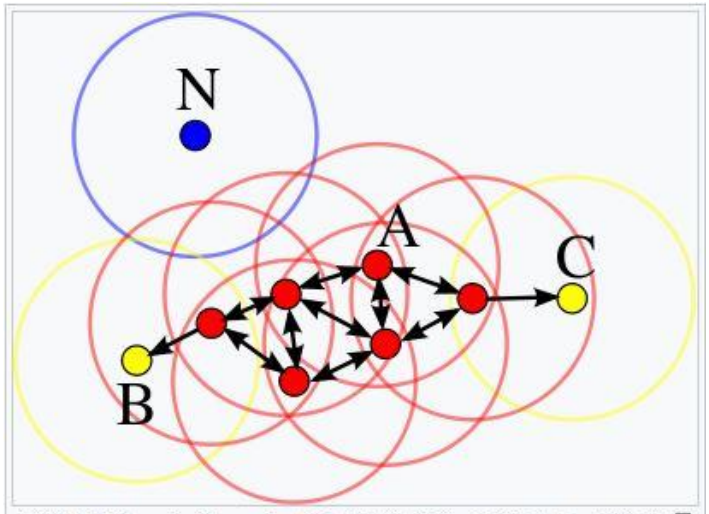
- 如果一个点 p 在距离 ε 范围内有至少 minPts 个点(包括自己)，则这个点被称为核心点，那些 ε 范围内的则被称为由 p 直接可达的。同时定义，没有任何点是由非核心点直接可达的。
- 如果存在一条道路 $p_1, ..., p_n$ ，有 $p_1 = p$ 和 $p_n = q$， 且每个 p_{i+1} 都是由 p_i 直接可达的(道路上除了 q 以外所有点都一定是核心点)，则称 q 是由 p 可达的。
- 所有不由任何点可达的点都被称为局外点。

如果 p 是核心点，则它与所有由它可达的点(包括核心点和非核心点)形成一个聚类，每个聚类拥有最少一个核心点，非核心点也可以是聚类的一部分，但它是在聚类的“边缘”位置，因为它不能达至更多的点。

“可达性”（英文：Reachability ）不是一个对称关系，因为根据定义，没有点是由非核心点可达的，但非核心点可以是由其他点可达的。所以为了正式地界定 DBSCAN 找出的聚类，进一步定义两点之间的“连结性”（英文：Connectedness） ：如果存在一个点 o 使得点 p 和点 q 都是由 o 可达的，则点 p 和点 q 被称为(密度)连结的，而连结性是一个对称关系。

定义了连结性之后，每个聚类都符合两个性质：

1. 一个聚类里的每两个点都是互相连结的；
2. 如果一个点 p 是由一个在聚类里的点 q 可达的，那么 p 也在 q 所属的聚类里。



在这幅图里，minPts = 4，点 A 和其他红色点是核心点，因为它们的 ε-邻域（图中红色圆圈）里包含最少 4 个点（包括自己），由于它们之间相互相可达，它们形成了一个聚类。点 B 和点 C 不是核心点，但它们可由 A 经其他核心点可达，所以也属于同一个聚类。点 N 是局外点，它既不是核心点，又不由其他点可达。

3. 伪代码

```
DBSCAN(D, eps, MinPts) {
   C = 0
   for each point P in dataset D {
      if P is visited
         continue next point
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else {
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)
      }
   }
}

expandCluster(P, NeighborPts, C, eps, MinPts) {
   add P to cluster C
   for each point P' in NeighborPts {
      if P' is not visited {
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
```

```
            NeighborPts = NeighborPts joined with NeighborPts'
        }
        if P' is not yet member of any cluster
           add P' to cluster C
     }
}

regionQuery(P, eps)
   return all points within P's eps-neighborhood (including
P)
```

二、 实验报告

1. 实验环境及数据

操作系统	Ubuntu 16.04 LTS
实验数据	long.csv
	moon.csv
	sizes5.csv
	spiral.csv
	square1.csv
	square4.csv

2. 实验结果