4.9 DBSCAN

CSDN学院

2017年11月

常用聚类算法

- 基于距离、相似度的聚类算法
  - *K*-means（*K*均值）及其变种（ *K*-centers 、 Mini Batch *K*-Means）
  - Mean shift
  - 吸引力传播（Affinity Propagation ，AP）
  - 层次聚类
    - 聚合聚类（Agglomerative Clustering）
- 基于密度的聚类算法
  - DBSCAN、DensityPeak（密度最大值聚类）
- 基于连接的聚类算法
  - 谱聚类

基于密度聚类方法

- 密度聚类方法：具有足够密度的区域划分为簇
  - 能克服基于距离的算法只能发现“类圆形”聚类的缺点，可发现任意形状的聚类，且对噪声数据不敏感。
  - 但计算密度过程的计算复杂度大，需要建立空间索引来降低计算量，且对数据维数的伸缩性较差。
  - 需要扫描整个数据库，每个数据对象都可能引起一次查询，因此当数据量大时会造成频繁的*I/O*操作。
  - 代表算法有：DBSCAN、OPTICS、DENCLUE算法等

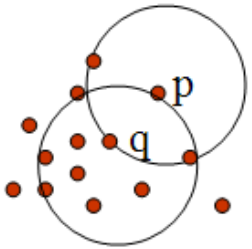
DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

- DBSCAN是一个有代表性的基于密度的聚类算法。
  - 簇：密度相连的点的最大集合
  - 可在有“噪声”的空间数据库中发现任意形状的聚类

基础概念

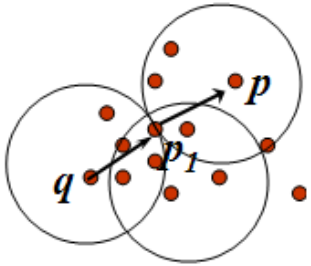
- DBSCAN的基本思想涉及的一些概念如下：
- (1) 对象的ε一邻域：给定对象的ε半径内的区域。
- (2) 核心点：一个对象的ε一邻域至少包含最小数目(MinPts)个对象，则称该对象为核心点。
- (3) 直接密度可达：给定一组对象集合D，如果*p*是在*q*的ε一邻域内，而*q*是一个核心点，则称对象*p*从对象*q*出发是直接密度可达的。



- ε=1，MinPts=5，*q*是一个核心对象
- 对象*p*从对象*q*出发是直接密度可达的

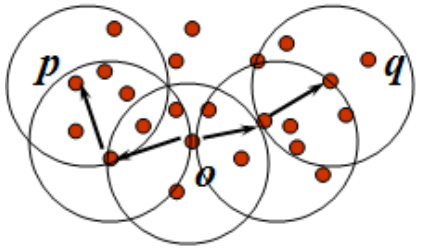
基础概念

- (4) 密度可达：如果存在一个对象链p_1，p_2，…，p_m，其中$p_1=p$，且$p_m=q$，对于$p_l \in D$，$(1 \le i \le n)$，p_{i+1}是从p_1关于ε和MinPts直接密度可达的，则对象p是从对象q关于ε和MinPts密度可达的。
- (5) 密度相连：如果对象集合D中存在一个对象o，使得对象p和q是从o关于ε和MinPts密度可达的，则对象p和q是关于ε和MinPts密度相连的。



直接密度可达

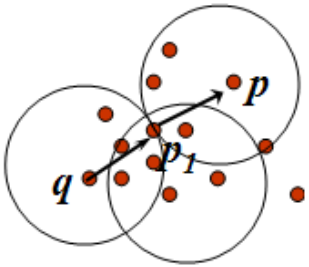
ε=1cm，MinPts=5，q是一个核心对象
$p1$是从q关于ε和MitPts直接密度可达
p是从p_1关于ε和MitPts直接密度可达，
则对象p从对象q关于ε和MitPts密度可达



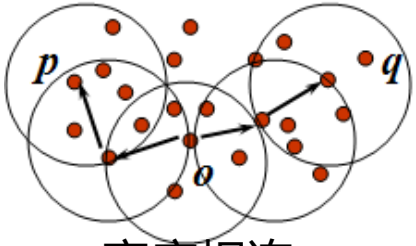
密度相连

基础概念

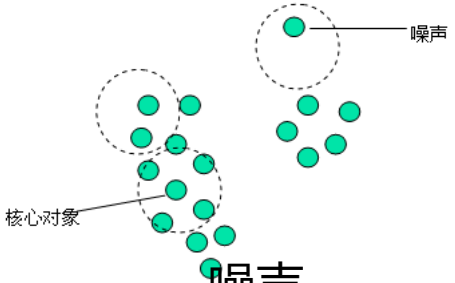
- (6) 边界点：非核心点，从某一核心点直接密度可达。
- (7) 噪声：聚类结束时，不属于任何簇的点。



直接密度可达



密度相连



噪声

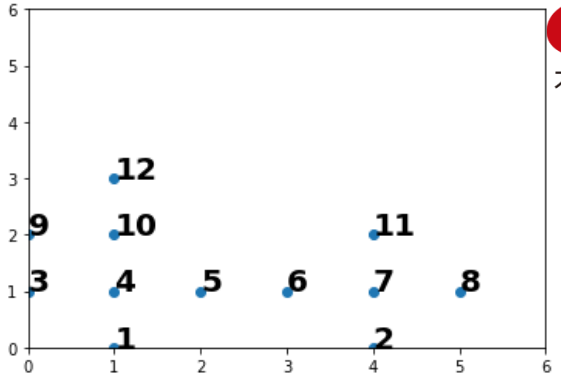
DBSCAN

- 参数：
  - 给定聚类对象的半径ε—邻域和
  - ε—邻域中最小包含的对象数MinPts
- 1. 检查某个对象ε—邻域中的对象数，如果对象数大于MinPts，该对象就是核心对象，就构建以该对象为核心的新簇。
- 2. 反复寻找从这些核心对象出发在ε—邻域内的对象，这个寻找过程可能会合并一些簇，直到没有新的对象可以添加到任何簇中为止。
  - 一个基于密度的簇是基于密度可达性的最大的密度相连对象的集合。
- 不包含在任何簇中的对象被认为是“噪声”。

DBSCAN算法描述

- 输入：包含N个对象的数据库，半径ε，最少数目MinPts。
- 输出：所有生成的簇，达到密度要求。
- 1. REPEAT
- 2. 从数据库中抽取一个未处理过的点；
- 3. IF 抽出的点是核心点 THEN找出所有从该点密度可达的对象，形成一个簇
- 4. ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一点；
- 5. UNTIL 所有点都被处理；

DBSCAN算法举例



- 训练数据如左所示，样本数N=12
- DBSCAN算法参数：ε=1，MinPts=4

序号	属性 1	属性 2
1	1	0
2	4	0
3	0	1
4	1	1
5	2	1
6	3	1
7	4	1
8	5	1
9	0	2
10	1	2
11	4	2
12	1	3

步骤	选择的点	在ε中点的个数	通过计算可达点而找到的新簇
1	1	2	无
2	2	2	无
3	3	3	无
4	4	5	簇C_1：{1，3，4，5，9，10，12}
5	5	3	已在一个簇C_1中
6	6	3	无
7	7	5	簇C_2：{2，6，7，8，11}
8	8	2	已在一个簇C_2中
9	9	3	已在一个簇C_1中
10	10	4	已在一个簇C_1中，
11	11	2	已在一个簇C_2中
12	12	2	已在一个簇C_1中

DBSCAN算法举例（cont.）

步骤	选择的点	在ε中点的个数	通过计算可达点而找到的新簇
1	1	2	无
2	2	2	无
3	3	3	无
4	4	5	簇C_1：{1，3，4，5，9，10，12}
5	5	3	已在一个簇C_1中
6	6	3	无
7	7	5	簇C_2：{2，6，7，8，11}
8	8	2	已在一个簇C_2中
9	9	3	已在一个簇C_1中
10	10	4	已在一个簇C_1中，
11	11	2	已在一个簇C_2中
12	12	2	已在一个簇C_1中

第1步，在数据库中选择一点1，由于在以它为圆心的，以1为半径的圆内包含2个点（小于4），因此它不是核心点，选择下一个点。

第2步，在数据库中选择一点2，由于在以它为圆心的，以1为半径的圆内包含2个点，因此它不是核心点，选择下一个点。

第3步，在数据库中选择一点3，由于在以它为圆心的，以1为半径的圆内包含3个点，因此它不是核心点，选择下一个点。

第4步，在数据库中选择一点4，由于在以它为圆心的，以1为半径的圆内包含5个点，因此它是核心点，寻找从它出发可达的点（直接可达4个，间接可达3个），聚出的新类{1，3，4，5，9，10，12}，选择下一个点。

第5步，在数据库中选择一点5，已经在簇1中，选择下一个点。

第6步，在数据库中选择一点6，由于在以它为圆心的，以1为半径的圆内包含3个点，因此它不是核心点，选择下一个点。

第7步，在数据库中选择一点7，由于在以它为圆心的，以1为半径的圆内包含5个点，因此它是核心点，寻找从它出发可达的点，聚出的新类{2，6，7，8，11}，选择下一个点。

第8步，在数据库中选择一点8，已经在簇2中，选择下一个点。

第9步，在数据库中选择一点9，已经在簇1中，选择下一个点。

第10步，在数据库中选择一点10，已经在簇1中，选择下一个点。

第11步，在数据库中选择一点11，已经在簇2中，选择下一个点。

第12步，选择12点，已经在簇1中，由于这已经是最后一点所有点都以处理，

Scikit learn中DBSCAN算法的实现

- *class* sklearn.cluster.DBSCAN(*eps=0.5, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=1*)

- **eps** 为ε，定于邻域大小
- **min_samples** 为MinPts，定于核心节点的条件，即邻域内样本点的最小数目
- **algorithm** 为最近邻的搜索算法，可为‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’

方法	参数	可扩展性	使用场景
K-Means	number of clusters	非常大的 n_samples, 中等的 n_clusters 使用 MiniBatch	通用, 均匀的簇大小 不是太多的簇
Affinity propagation	damping, sample preference	n_samples 不可扩展	许多簇，不均匀的簇大小
Mean-shift	bandwidth	n_samples不可扩展	许多簇，不均匀的簇大小
Spectral clustering	number of clusters	中等的n_samples, 小的 n_clusters	很少簇，均匀的簇大小
Ward hierarchical clustering	number of clusters	大的 n_samples 和 n_clusters	很多的簇，可以增加连接限制
Agglomerative clustering	number of clusters, linkage type, distance	大的 n_samples 和 n_clusters	很多簇，可能连接限制，非欧几里得距离
DBSCAN	neighborhood size	非常大的 n_samples, 中等的 n_clusters	不均匀的簇大小
Gaussian mixtures（高斯混合）	many	不可扩展	适用于密度估计
Birch	branching factor，threshold, optional global clusterer	大的 n_clusters 和 n_samples	大数据集，异常值去除，数据简化

更多信息请见sklearn用户手册： http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

CSDN
不 止 于 代 码

THANK YOU

AI100