刘建平Pinard

十年码农，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

DBSCAN密度聚类算法

DBSCAN(Density-Based Spatial Clustering of Applications with Noise，具有噪声的基于密度的聚类方法)是一种很典型的密度聚类算法，和K-Means，BIRCH这些一般只适用于凸样本集的聚类相比，DBSCAN既可以适用于凸样本集，也可以适用于非凸样本集。下面我们就对DBSCAN算法的原理做一个总结。

1. 密度聚类原理

DBSCAN是一种基于密度的聚类算法，这类密度聚类算法一般假定类别可以通过样本分布的紧密程度决定。同一类别的样本，他们之间的紧密相连的，也就是说，在该类别任意样本周围不远处一定有同类别的样本存在。

通过将紧密相连的样本划为一类，这样就得到了一个聚类类别。通过将所有各组紧密相连的样本划为各个不同的类别，则我们就得到了最终的所有聚类类别结果。

2. DBSCAN密度定义

在上一节我们定性描述了密度聚类的基本思想，本节我们就看看DBSCAN是如何描述密度聚类的。DBSCAN是基于一组邻域来描述样本集的紧密程度的，参数(ϵ, MinPts)用来描述邻域的样本分布紧密程度。其中，ϵ描述了某一样本的邻域距离阈值，MinPts描述了某一样本的距离为ϵ的邻域中样本个数的阈值。

假设我的样本集是D=$(x_1, x_2, \ldots, x_m)$,则DBSCAN具体的密度描述定义如下：

1) ϵ-邻域：对于$x_j \in D$，其ϵ-邻域包含样本集D中与x_j的距离不大于ϵ的子样本集，即$N_\epsilon(x_j) = \{x_i \in D | distance(x_i, x_j) \leq \epsilon\}$, 这个子样本集的个数记为$|N_\epsilon(x_j)|$

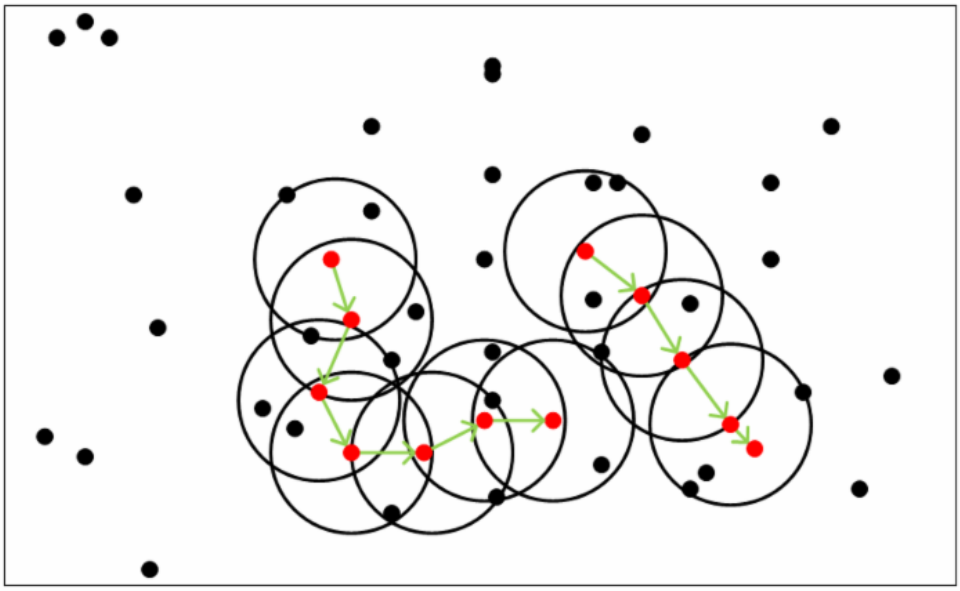
2) 核心对象：对于任一样本$x_j \in D$，如果其ϵ-邻域对应的$N_\epsilon(x_j)$至少包含MinPts个样本，即如果$|N_\epsilon(x_j)| \geq MinPts$，则$x_j$是核心对象。

3) 密度直达：如果x_i位于x_j的ϵ-邻域中，且x_j是核心对象，则称x_i由x_j密度直达。注意反之不一定成立，即此时不能说x_j由x_i密度直达, 除非且x_i也是核心对象。

4) 密度可达：对于x_i和x_j,如果存在样本样本序列$p_1, p_2, \ldots, p_T$满足$p_1 = x_i, p_T = x_j$, 且p_{t+1}由p_t密度直达，则称x_j由x_i密度可达。也就是说，密度可达满足传递性。此时序列中的传递样本$p_1, p_2, \ldots, p_{T-1}$均为核心对象，因为只有核心对象才能使其他样本密度直达。注意密度可达也不满足对称性，这个可以由密度直达的不对称性得出。

5) 密度相连：对于x_i和x_j,如果存在核心对象样本x_k，使x_i和x_j均由x_k密度可达，则称x_i和x_j密度相连。注意密度相连关系是满足对称性的。

从下图可以很容易看出理解上述定义，图中MinPts=5，红色的点都是核心对象，因为其ϵ-邻域至少有5个样本。黑色的样本是非核心对象。所有核心对象密度直达的样本在以红色核心对象为中心的超球体内，如果不在超球体内，则不能密度直达。图中用绿色箭头连起来的核心对象组成了密度可达的样本序列。在这些密度可达的样本序列的ϵ-邻域内所有的样本相互都是密度相连的。



有了上述定义，DBSCAN的聚类定义就简单了。

3. DBSCAN密度聚类思想

DBSCAN的聚类定义很简单：由密度可达关系导出的最大密度相连的样本集合，即为我们最终聚类的一个类别，或者说一个簇。

这个DBSCAN的簇里面可以有一个或者多个核心对象。如果只有一个核心对象，则簇里其他的非核心对象样本都在这个核心对象的ϵ-邻域里；如果有多个核心对象，则簇里的任意一个核心对象的ϵ-邻域中一定有一个其他的核心对象，否则这两个核心对象无法密度可达。这些核心对象的ϵ-邻域里所有的样本的集合组成的一个DBSCAN聚类簇。

那么怎么才能找到这样的簇样本集合呢？DBSCAN使用的方法很简单，它任意选择一个没有类别的核心对象作为种子，然后找到所有这个核心对象能够密度可达的样本集合，即为一个聚类簇。接着继续选择另一个没有类别的核心对象去寻找密度可达的样本集合，这样就得到另一个聚类簇。一直运行到所有核心对象都有类别为止。

基本上这就是DBSCAN算法的主要内容了，是不是很简单？但是我们还是有三个问题没有考虑。

第一个是一些异常样本点或者说少量游离于簇外的样本点，这些点不在任何一个核心对象在周围，在DBSCAN中，我们一般将这些样本点标记为噪音点。

第二个是距离的度量问题，即如何计算某样本和核心对象样本的距离。在DBSCAN中，一般采用最近邻思想，采用某一种距离度量来衡量样本距离，比如欧式距离。这和KNN分类算法的最近邻思想完全相同。对应少量的样本，寻找最近邻可以直接去计算所有样本的距离，如果样本量较大，则一般采用KD树或者球树来快速的搜索最近邻。如果大家对于最近邻的思想，距离度量，KD树和球树不熟悉，建议参考之前写的另一篇文章K近邻法(KNN)原理小结。

第三种问题比较特殊，某些样本可能到两个核心对象的距离都小于ϵ，但是这两个核心对象由于不是密度直达，又不属于同一个聚类簇，那么如果界定这个样本的类别呢？一般来说，此时DBSCAN采用先来后到，先进行聚类的类别簇会标记这个样本为它的类别。也就是说BDSCAN的算法不是完全稳定的算法。

4. DBSCAN聚类算法

下面我们对DBSCAN聚类算法的流程做一个总结。

输入：样本集D=$(x_1, x_2, \ldots, x_m)$，邻域参数$(\epsilon, MinPts)$，样本距离度量方式

输出： 簇划分C.

1）初始化核心对象集合$\Omega = \emptyset$，初始化聚类簇数k=0，初始化未访问样本集合Γ = D，簇划分C = $\emptyset$

2）对于j=1,2,...m，按下面的步骤找出所有的核心对象：

a）通过距离度量方式，找到样本x_j的ϵ-邻域子样本集$N_\epsilon(x_j)$

b）如果子样本集样本个数满足$|N_\epsilon(x_j)| \geq MinPts$， 将样本$x_j$加入核心对象样本集合：$\Omega = \Omega \cup \{x_j\}$

3）如果核心对象集合$\Omega = \emptyset$，则算法结束，否则转入步骤4.

4）在核心对象集合Ω中，随机选择一个核心对象o，初始化当前簇核心对象队列$\Omega_{cur} = \{o\}$，初始化类别序号k=k+1，初始化当前簇样本集合$C_k = \{o\}$，更新未访问样本集合$\Gamma = \Gamma - \{o\}$

5）如果当前簇核心对象队列$\Omega_{cur} = \emptyset$，则当前聚类簇C_k生成完毕，更新簇划分C=$\{C_1, C_2, \ldots, C_k\}$，更新核心对象集合$\Omega = \Omega - C_k$， 转入步骤3。

6）在当前簇核心对象队列Ω_{cur}中取出一个核心对象o'，通过邻域距离阈值ϵ找出所有的ϵ-邻域子样本集$N_\epsilon(o')$，令$\Delta = N_\epsilon(o') \cap \Gamma$，更新当前簇样本集合$C_k = C_k \cup \Delta$，更新未访问样本集合$\Gamma = \Gamma - \Delta$，更新$\Omega_{cur} = \Omega_{cur} \cup (N_\epsilon(o') \cap \Omega) - o'$，转入步骤5.

2. 梯度提升树(GBDT)原理小结(86910)
3. 线性判别分析LDA原理总结(61993)
4. scikit-learn决策树算法类库使用小结(44319)
5. word2vec原理(一) CBOW与Skip-Gram模型基础(43832)

评论排行榜

1. 梯度提升树(GBDT)原理小结(172)
2. 集成学习之Adaboost算法原理小结(109)
3. 谱聚类（spectral clustering）原理总结(98)
4. 梯度下降（Gradient Descent）小结(97)
5. 线性判别分析LDA原理总结(75)

推荐排行榜

1. 梯度下降（Gradient Descent）小结(56)
2. 奇异值分解(SVD)原理与在降维中的应用(28)
3. 卷积神经网络(CNN)反向传播算法(18)
4. 集成学习之Adaboost算法原理小结(18)
5. 集成学习原理小结(17)

输出结果为： 簇划分C=$\{C_1, C_2, \ldots, C_k\}$

5. DBSCAN小结

和传统的K-Means算法相比，DBSCAN最大的不同就是不需要输入类别数k，当然它最大的优势是可以发现任意形状的聚类簇，而不是像K-Means，一般仅仅使用于凸的样本集聚类。同时它在聚类的同时还可以找出异常点，这点和BIRCH算法类似。

那么我们什么时候需要用DBSCAN来聚类呢？一般来说，如果数据集是稠密的，并且数据集不是凸的，那么用DBSCAN会比K-Means聚类效果好很多。如果数据集不是稠密的，则不推荐用DBSCAN来聚类。

下面对DBSCAN算法的优缺点做一个总结。

DBSCAN的主要优点有：

1） 可以对任意形状的稠密数据集进行聚类，相对的，K-Means之类的聚类算法一般只适用于凸数据集。

2） 可以在聚类的同时发现异常点，对数据集中的异常点不敏感。

3） 聚类结果没有偏倚，相对的，K-Means之类的聚类算法初始值对聚类结果有很大影响。

DBSCAN的主要缺点有：

1）如果样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差，这时用DBSCAN聚类一般不适合。

2） 如果样本集较大时，聚类收敛时间较长，此时可以对搜索最近邻时建立的KD树或者球树进行规模限制来改进。

3） 调参相对于传统的K-Means之类的聚类算法稍复杂，主要需要对距离阈值ϵ，邻域样本数阈值MinPts联合调参，不同的参数组合对最后的聚类效果有较大影响。

分类: 0081. 机器学习

标签： 聚类算法

刘建平Pinard
关注 - 14
粉丝 - 1940
+加关注

« 上一篇：用scikit-learn学习BIRCH聚类
» 下一篇：用scikit-learn学习DBSCAN聚类

posted @ 2016-12-22 16:32 刘建平Pinard 阅读(32980) 评论(19) 编辑 收藏

评论列表

#1楼 2016-12-22 17:56 FIGHTING360

文章挺好，给个建议，文章太偏学术化了，本人学术搞了3年，也发了几篇EI论文，但觉得国内大多数文章好水。建议多搞点实在的，比如可以将原理用代码来实现，大家一同探讨，而不是简简单单的将纯学术的东西弄一遍，东西不是越复杂越好。

支持(4) 反对(10)

#2楼[楼主] 2016-12-22 18:03 刘建平Pinard

@ 649727360
谢谢你的建议，一般我对每个算法都有原理篇和实践篇，这篇是原理篇，所以可能偏学术一些。至于代码实现，由于大段代码在文章中并不适合阅读，且工作闲余时间不是那么多，因此没有放自己写的代码。以后会考虑，总之，非常感谢你的建议。

支持(11) 反对(0)

#3楼 2017-03-30 15:51 Allen_xiaoshi

”在当前簇核心对象队列ΩcurΩcur中取出一个核心对象o“，这句话我这么理解：就是在以选中的核心对象多包含的点中再找一个核心对象，然后重复下去。这样一个一个核心对象就构成了核心对象队列

这样理解可以吗

支持(0) 反对(0)

#4楼[楼主] 2017-03-30 16:07 刘建平Pinard

@ Allen_xiaoshi
你好，基本正确。但是会有某一些核心对象由于被吸收到前面聚类的核心对象中，而没有机会被单独取出来。

支持(0) 反对(0)

#5楼 2017-05-02 09:59 feng清扬

能写一篇怎么选择eps和MinPts的文章吗，我觉得这才是密度聚类的最难的地方。

支持(1) 反对(0)

#6楼[楼主] 2017-05-02 10:22 刘建平Pinard

@ feng清扬
你好，我的这篇文章可能可以部分回答你的问题：http://www.cnblogs.com/pinard/p/6217852.html。
至于具体如何选择，个人选择时一般会考虑要分析的数据各特征的方差和均值的分布，也会考虑数据领域的一些先验知识。

支持(1) 反对(0)

#7楼 2017-10-09 20:45 董浩Razor

2） 可以在聚类的同时发现异常点，对数据集中的异常点不敏感。
是不敏感吗？我觉得挺敏感的。。~是不是我理解错了

支持(0) 反对(0)

#8楼[楼主] 2017-10-10 11:05 刘建平Pinard

@ 董浩Razor
这里异常点不敏感的原因主要是大多数异常点都是离群点。

由于我们有邻域参数的限制，会让这些异常点自己独立成为单独的一个簇。因此样本数极少的簇我们可以认定为是异常点将其排除。当然也有一些很特殊的情况，比如邻域参数选择的原因，导致异常点和正常点的距离不是足够远，可能会导致判断错误，将其加入正常簇。

但大多数时候，只要邻域参数选择适当，是可以排除异常点的。

支持(0) 反对(0)

#9楼 2017-12-27 21:31 朗月白首

请问下当前簇核心对象队列Ωcur是怎么更新的？算法中那个队列初始化一个对象之后就没有新对象加进去了，这样步骤5取出初始化的那个核心对象没有了。

支持(0) 反对(0)

#10楼[楼主] 2017-12-28 10:56 刘建平Pinard

@ 朗月白首
你好，参看算法步骤2.b)

支持(0) 反对(0)

#11楼 2017-12-28 11:18 朗月白首

@ 刘建平Pinard
算法步骤2.b)是先把所有的核心对象找出来并放在Ω中。 我是想问下当前核心对象队列Ωcur是怎么更新的。

支持(0) 反对(0)

#12楼[楼主] 2017-12-28 11:35 刘建平Pinard

@ 朗月白首
你好，这里我自己总结的时候写漏了，已经补上，感谢指出错误。

支持(0) 反对(0)

#13楼 2018-05-17 09:36 kylin0228

您好 在第三节的您提出的第三个问题中：由于两个核心对象不是密度直达这里
是不是应该改成密度可达？？

支持(0) 反对(0)

#14楼[楼主] 2018-05-17 17:11 刘建平Pinard

@ kylin0228
你好，密度直达是强关系，这样的核心对象肯定会在一个簇的。不会分开。因此不会出现上面的第三种情况。
而密度可达则就不一定了，有可能因为选择核心对象的的随机先后顺序而被分到不同的簇里面。

支持(0) 反对(0)

#15楼 2018-05-24 11:38 seekermp4

刘老师 请问如何定义凸样本集，百度了下没有找到相关定义，谢谢~

支持(0) 反对(0)

#16楼[楼主] 2018-05-24 23:04 刘建平Pinard

@ seekermp4
你好，凸集是凸优化里面的概念，如果你要完全搞清楚，需要看<凸优化>这本书。

简单点说集合里任意两点连接成的线段中的点也在集合中，那么就是凸集。

支持(1) 反对(0)

#17楼 2018-06-06 10:57 seekermp4

@ 刘建平Pinard
谢谢刘老师的耐心解答 明白了！

支持(0) 反对(0)

#18楼 2018-07-18 10:27 grace甜

老师您好：
“Ωcur=Ωcur∪(Nε(o′)∩Ω)，转入步骤5.”

Ωcur是如何更新成Ωcur=ø的呢？6中的并集，只会越并越多吧？ 至少也有一个o′。

希望老师指点。

支持(0) 反对(0)

#19楼[楼主] 2018-07-19 10:08 刘建平Pinard

@ grace甜
你好，这里可能写的不严谨，其实是在第6步开始的时候，我们会从Ω_{cur}中取出一个核心对象o',此时隐含着
$\Omega_{cur} = \Omega_{cur} - o'$

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部