

机器学习Homework 5——聚类分析实验报告

第1节 DBSCAN的算法框架

算法 1 随机森林算法

输入: 训练样本集 $D = \{(x_i)\}_{i=1}^N, x_i \in X, i = 1, 2, \dots, N$; 邻域参数 $(\epsilon, MinPts)$

输出: 样本簇划分 $C = \{C_k\}$ (簇的个数依赖于邻域参数的选择)

1. 初始化核心对象集合 $\Omega = \emptyset$, 初始化簇的类别 $C_0, k = 0$ 。
 2. 遍历 D 的元素, 如果当前遍历的元素是核心对象, 则将其加入核心对象集合 Ω 当中, 如果不是核心对象, 则将其标记为噪声。
 3. 在核心对象集合 Ω 中, 随机选择一个未访问的核心对象 o , 首先将 o 标记为已访问, 然后将 o 标记类别 C_0 , 最后将 o 的 ϵ -邻域中未访问的数据, 存放到种子集合 $seeds$ 中。
 4. 如果种子集合 $seeds = \emptyset$, 则当前聚类簇 C_k 生成完毕, 且 $k = k + 1$, 跳转到步5, 否则, 从种子集合 $seeds$ 中挑选一个种子点 s , 首先将其标记为已访问, 标记类簇 C_k , 然后判断 s 是否为核心对象, 如果 s 是核心对象, 则将 s 的 ϵ -邻域中未访问的数据加入到种子集合 $seeds$ 中, 跳转步4。
 5. 如果核心对象集合 Ω 中元素都已经被访问, 则算法结束, 否则转入步骤3。
-

第2节 DBSCAN算法的实验分析

本节实验旨在对DBSCAN算法进行实验分析, 我使用python语言的sklearn机器学习软件包进行实验。

第2.1小节 数据集简介

为方便可视化分析, 我选用了sklearn中两个数据生成函数来生成数据进行实验:

1. `make_blobs`: 生成各向同性高斯点用于聚类，设定四个均值中心，即真实的类别应该有4个。
2. `make_circles`: 生成在二维空间中生成一个大圆包含着一个小圆，即真实的类别应该有2个。

用以上两个函数分别生成750条数据用于聚类实验分析。

第2.2小节 DBSCAN算法与其他聚类算法实验对比

我使用K-Means聚类算法和高斯混合模型(GMM)与DBSCAN算法进行对比，其中DBSCAN算法默认设定 ϵ 为0.3， $MinPts$ 为15，最终各聚类算法的表现如下表1、2所示：

表 1: 不同聚类算法在`make_blobs`生成数据上的表现

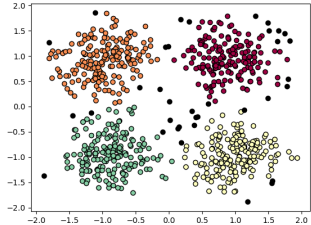
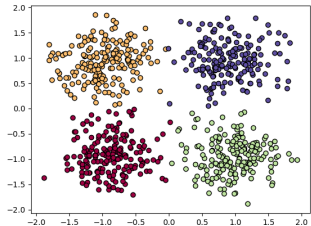
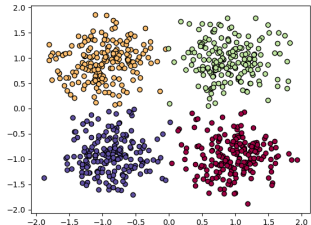
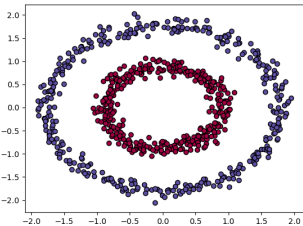
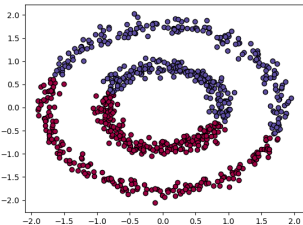
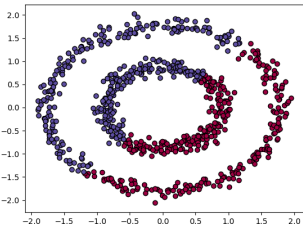
聚类算法	可视化图片	轮廓系数	V-measure	噪声点
DBSCAN		0.575	0.865	36
K-Means(k=4)		0.612	0.918	0
GMM(k=4)		0.612	0.918	0

表 2: 不同聚类算法在make_circles生成数据上的表现

聚类算法	可视化图片	轮廓系数	V-measure	噪声点
DBSCAN		0.113	1.0	0
K-Means(k=2)		0.354	0.0	0
GMM(k=2)		0.352	0.0	0

首先对于以四个均值中心生成的各向同性高斯点数据而言，虽然从量化指标上看，轮廓系数和V-measure都不如预先设定簇数为4的K-Means和GMM算法，但从可视化图片上观察，DBSCAN和k为4的K-Means、GMM算法均能够很好地将数据聚成4个簇。另外，由于DBSCAN其算法原理中存在将非核心对象结点标记为噪声点的一个步骤，对于make_blobs生成的数据，DBSCAN算法的聚类结果存在36个噪声点，在可视化图片中使用黑色标注出，可以发现，这些噪声点都是处于离各个簇的簇心都很远的位置。

另外，在make_circles生成的数据上DBSCAN则表现出了其作为基于密度的聚类算法的优越性，如表2所示，DBSCAN将内外两个圆分别正确划分成两个类别，而K-Means、GMM算法都形成了错误的超平面，将两个圆进行了错误的划分，这一点从V-measure指标中也可以明显看出，V-measure作为聚类结果同质性和完整性的调和均值，DBSCAN达到了1.0这个完美的分数。然而，如表2所示，在轮廓系数指标上，正确聚类的DBSCAN依然不如错误聚类的K-Means和GMM，经过查阅资料之后我发现，对于簇结构为凸的数据轮廓系数值高，而对于簇结构非凸需要使用DBSCAN进行聚类的数据，轮廓系数值低，因此，轮廓系数在make_circles生成的非凸的结构不能正确描述DBSCAN与其他算法的对比。

总体而言，DBSCAN可以对任意形状的稠密数据集进行聚类，相对的，K-Means、GMM等聚类算法一般只适用于凸数据集；另外DBSCAN算法可以在聚类时发现噪声点，对数据集中的异常点不敏感；最后DBSCAN对初始值不敏感，不需要预先设定簇数 k ，相对的，正如下一节内容所分析的那样，K-Means、GMM等聚类算法 k 值的选取对聚类结果有很大影响。

第3节 关于簇数 k 对聚类算法影响的讨论

本节实验中，我选取了K-Means算法来研究簇数 k 对聚类结果的影响，并确定出 k 值的确定策略。对此，我对K-Means算法选取了不同的 k 值，得到 k 值与聚类簇的平均直径的关系图1和不同 k 值在可视化图片以及量化指标的表格3。

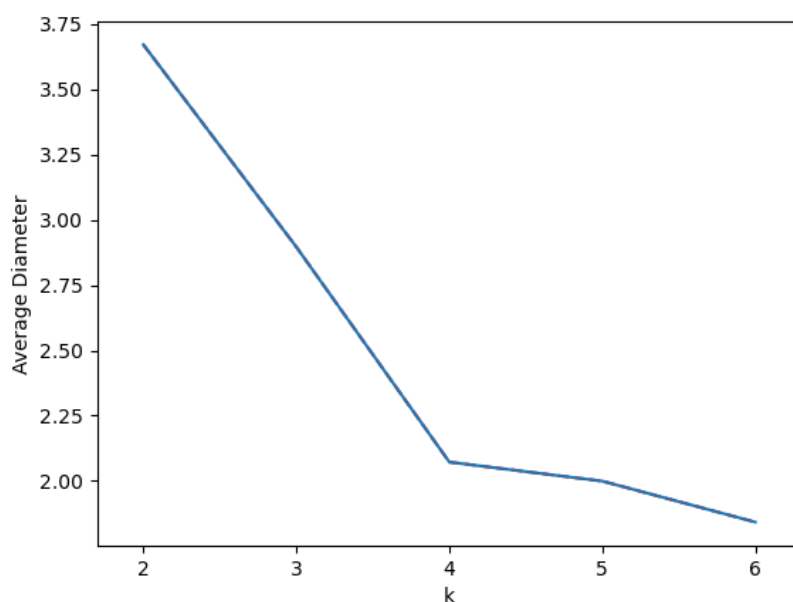
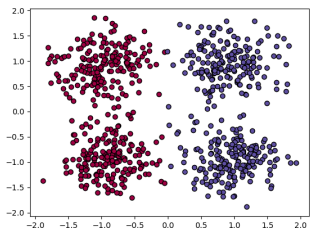
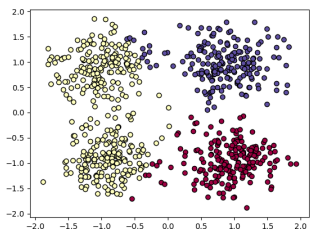
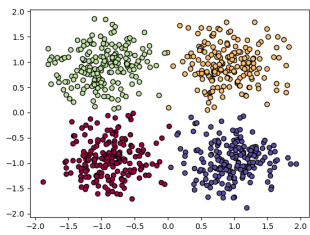
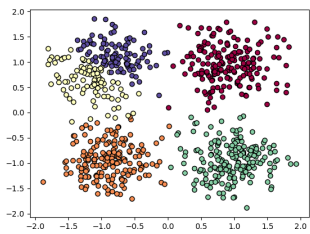
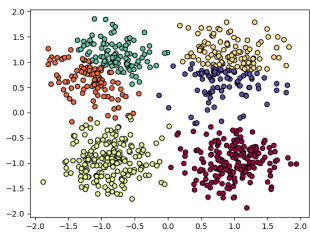


图 1: k 值与平均直径的关系

表 3: 不同k值对于K-Means算法的影响

k	可视化图片	轮廓系数	V-measure	平均直径
2		0.434	0.605	3.671
3		0.484	0.746	2.896
4		0.612	0.918	2.072
5		0.535	0.862	1.999
6		0.454	0.803	1.842

从表3中可以看出, 随着k值的选择越接近数据本身正确的簇数, 可视化结果和量化指标的表现越好。另一方面, 从图1可以看出, 只要我预先设定的类簇的k等于或者高于真实的类簇的数目时, 随着k的减小平均直径上升会很缓慢, 而一旦k值少于真实数目的类簇时, 该指标就会急剧上升, 即k值与聚类簇的平均直径关系图中的拐点的k值即为最接近真实的类簇的数目的取值, 在本实验情况下, 该取值为4。因此在实验中可以采用二分查找的策略快速找到最优的k值。

第4节 总结

在本作业中，我首先给出了DBSCAN完整的算法框架，并生成了两类数据，分别对比DBSCAN算法和K-Means、GMM算法在凸数据和非凸数据下的聚类表现，由此展现了DBSCAN在对非凸结构数据进行聚类时的优越性。另外在我继续研究了预先设定的簇数 k 对K-means算法的影响，确定了依据聚类簇平均直径通过二分查找快速确定最优 k 值的策略。