|  |  |
| --- | --- |
| kNN检索速度测试报告 | 文件类型：测试文档 |
| 文件编号： |
| 面向的部门： 研发中心 |
| 保密等级：高 |
| 作者： 刘凯 |
| 日期：2018-08-30 |
| 版本：1.0V |

kNN检索速度测试报告



目 录

[1 kNN算法解析 1](#_Toc523402054)

[2 kNN检索速度测试 1](#_Toc523402055)

# kNN算法解析

kNN(k-Nearest Neighbors)是一种基本的分类和回归算法，其算法原理为选取训练样本中与测试数据距离最近的个样本，将测试数据划分为个邻居中比重最大的一类。具体而言，设为训练集合，其中为维的特征向量，为其标号。对于测试数据，计算，其中为距离计算函数，可以为欧几里得距离、曼哈顿距离和余弦距离等。选取最小的个距离及其对应的标号，其中类别比重最大的一类就是测试数据的分类。kNN的主要参数是的选取，一般取值较小，在20以内，值在一定程度上会影响最后的分类结果，如图1所示。一般用交叉验证来确定值。

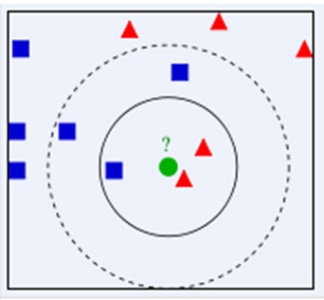


图1：k值对kNN分类的影响

# kNN检索速度测试

本文对两种kNN开源库进行了百万级特征库的检索速度测试，测试的库为pyflann和sklearn。其中pyflann为flann(Fast Library for Approximate Nearest Neighbors)的python接口，其计算内核还是FLANN的c++库；sklearn为python的一个机器学习库，包含了kNN的实现。算法的特征库维度为，其中为特征库的大小，为每个特征的维度。特征库分成100个类(class)，分别在这100个区间随机生成，每个区间生成10000个样本。特征标号分别为0~99。测试数据同样为2048维的随机数。最近邻数。

测试电脑配置为：intel i7-7700处理器，3.6GHz，16G内存，64位windows10操作系统。对不同测试数据数量的检索时间结果如表1所示：

表1：kNN检索速度结果(单位：s)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Pyflann | 0.81 | 7.99 | 85.39 |
| Sklearn | 4.42 | 4.94 | 6.67 |

说明1，上述结果均采用遍历的方法，由于不同类的特征库之间没有交叉，上述方法的分类结果均正确。pyflann的结果包括读取数据和计算距离，输出为5个最近的距离和索引，未进行标签分类，检索的时间和测试数据数量呈线性关系。而sklearn的结果包括训练，计算距离和标签分类，可以看到在时，所用的时间要比pyflann慢，但是当时，所用的时间远远小于pyflann。其原因应该是pyflann内部进行过优化，具体的优化算法暂未详细研究。但从上面的结果可以看出，在PC上进行kNN检索百万级特征库的时间应该在1s的数量级。

说明2，在调试过程中出现的问题。一开始产生一百万个特征库时，检索速度很慢，但我将特征库数量减小到80万，50万，10万和1万时，则检索速度迅速提升，且检索时间随着特征库数量线性地减小。推测原因是一百万个特征库的数据超出了电脑缓存空间的大小。后面将特征库的数据格式由‘float64’改为‘flaot32’后，检索时间缩短，且与前面的结果保持线性。