**标定文件格式**

标定文件，又称groundtruth文件，一般用来做误差量化，最终用于验证分类算法的准确性。

data/SIFT1B/gnd/idx\_1000M.ivecs

data/DEEP1B/deep1B\_groundtruth.ivecs

分别是sift1b和deep1b完整数据集的groundtruth文件

下面是不同数据集下的标定文件，可以看出sift1b针对多种数据集都提供了标定文件，deep1b只有针对完整10亿条记录的标定数据

DEEP1B/deep1B\_groundtruth.ivecs

SIFT1B/sift/sift\_groundtruth.ivecs

SIFT1B/siftsmall/siftsmall\_groundtruth.ivecs

SIFT1B/gnd/idx\_2M.ivecs

SIFT1B/gnd/idx\_200M.ivecs

SIFT1B/gnd/idx\_5M.ivecs

SIFT1B/gnd/idx\_1M.ivecs

SIFT1B/gnd/idx\_500M.ivecs

SIFT1B/gnd/idx\_20M.ivecs

SIFT1B/gnd/idx\_100M.ivecs

SIFT1B/gnd/idx\_50M.ivecs

SIFT1B/gnd/idx\_10M.ivecs

SIFT1B/gnd/idx\_1000M.ivecs

SIFT1B/gist/gist\_groundtruth.ivecs

标定文件不能由代码生成，需要数据提供者提供

标定文件读取通过定义在 utils.h 里面的函数

**template**<**typename** **T**>

**void** **readXvec**(std::ifstream &in, **T** \*data, **const** size\_t d, **const** size\_t n = 1)

和索引文件加载不同，groundtruth文件加载没有封装成基于路径的加载函数，而是自己打开文件，然后通过上面的函数加载数据

从代码可以看出，groundtruth文件格式非常简单，以向量为单位逐条存放：

|  |
| --- |
| 向量记录 1 |
| 向量记录 1 |
| ... |
| 向量记录 N |

从文件格式设计上看，向量记录可以不是等长度的，因为每条记录的格式如下：

|  |  |
| --- | --- |
| 当前记录对应的向量维度 N（4 字节，uint32\_t类型） | 向量数据（N 维，类型 idx\_t，对sift1b和deep1b而言，其实就是uint32\_t类型） |

groundtruth的维度指的是，每个查询对应的groundtruth邻节点数，这个由groundtruth文件的提供者提供，从代码看，如果传递给 **readXvec** 函数的维度和从groundtruth文件读取出来的维度不匹配，会导致程序异常退出。从 ivf-hnsw 里面提供的测试脚本看，deep1b采用维度值是1，sift1b采用的维度值是1000。

**readXvec** 函数的最后一个参数是查询数，代表后面搜索算法同时执行的查询数

从KNN相关的论文看，一般用查询数评估算法的扩展性，包括查询数和CPU时间的关系，查询数和I/O数的关系

|  |  |
| --- | --- |
| 组成 | 说明 |
| 向量维度 | uint32\_t 类型，4 字节 |
| 向量索引 | 每个 centriod，需要一个向量描述当前 centriod 里面各个点的索引值，这里的向量维度就是centriod里面的点数  这里包含了所有的centriod的索引  每个索引当作一条记录  这些记录顺序存放  每条记录的格式都是：  4 字节的长度信息，表示 centriod 里面的点数  和长度信息匹配的每个点的索引值（32bit整数构成的向量，每一维对应一个点） |

TODO:

根据这里的描述，编写标定文件验证工具，如果发现不符合上面的描述，报告错误，以便修正本文档，或者修正验证工具