**PQ量化文件格式**

PQ 文件的扩展名为：".pq" ，PQ文件不是sift1b和deep1b自带的，而是通过对训练数据作训练产生的

例如，运行完 bash examples/run\_deep1b.sh 后，会生成：

models/DEEP1B/norm\_pq16.pq

models/DEEP1B/pq16.pq

生成PQ文件的过程是：

1、加载学习向量集

data/DEEP1B/deep1B\_learn.fvecs

2、从上面的向量数据中随机选取一定数量的向量作为学习向量

3、通过IndexIVF\_HNSW.cpp定义的如下函数

**void** **IndexIVF\_HNSW::train\_pq**(size\_t n, **const** **float** \*x)

得到PQ训练结果

4、通过faiss/index\_io.cpp定义的如下函数

**void** **write\_ProductQuantizer** (**const** ProductQuantizer\*pq, **const** **char** \*fname)

将训练结果写入文件

从训练结果写入代码可以看出，PQ文件格式如下：

|  |  |
| --- | --- |
| 组成 | 说明 |
| 输入向量大小，也就是向量的维度  这是 ProductQuantizer 的构造函数的参数d，这个构造函数在 IndexIVF\_HNSW 的构造函数会被调用，初始化d的参数是dim | size\_t 类型，8 字节 |
| 子量化器数目 | size\_t 类型，8 字节 |
| 每个量化器的索引精度（比特数） | size\_t 类型，8 字节 |
| 向量总长度，就是随后的所有向量数据的总字节数 | size\_t 类型，8 字节 |
| 向量记录 1 | 每条记录的长度就是 sizeof(float) x 向量维度 |
| 向量记录 2 |  |
| ....... |  |
| 向量记录 N |  |

TODO:

根据这里的描述，编写PQ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，验证的核心就是根据向量维度和向量总长度，看后面向量记录数是否匹配