**作业二实验记录**

* **实验要求**

在上次的数据集上实现一个朴素贝叶斯分类器，并在训练集和测试集上分别进行测试。

* **实验步骤**

首先，根据上次划分好的training set和test set，目标是在每一个数据集下构建一个贝叶斯分类器。（这些数据集的groundtruth是20个类，编号依次为0-19，根据随机分类，训练集共有15056个文件，测试集共有3772个文件）

下面以training set为例，进行说明。

首先要拿到所有doc的词频，这个在上次构建VSM的时候，已经拿到了，记为tf\_dict.pkl，因此可以直接拿来使用。然后对于training set中的每一个doc，通过查找总文件拿到每一个doc的词频信息以及类别信息，存入train\_tf.pkl中。我们的目标是去计算给定一篇文档，他属于某个类的概率值大小。选择概率最大的作为预分类的一个类，再去计算准确率的大小。根据贝叶斯公式，我们就需要转化为计算词频先验和类别先验。针对词频先验，计算的是当前文档中的词频与每个类别下面的所有词频的比值，针对类别先验，计算的是每个类下的文档数之和占总体文档数的比值。

在这个过程中，我们可以用到3种模型，伯努利模型、多项式模型和混合模型。其中，伯努利模型只考虑到了词项的出现与否，而不考虑频率，计算速度会快，但是准确率不高。多项式模型不仅仅要考虑到当前评测doc中的词频，也要考虑到这些出现了的词语在所有文本中的词频，准确率相较会高一点。混合模型在当前文档中的词频不予考虑，只考虑到了总体的词频。在本次试验中，我采用的是多项式模型。虽然这个实现起来比前两种要更繁琐一些，但是准确率得到了很大的提高。

* **实验结果以及说明**

1. 由于上次对词典的删减力度较大，导致后来在做KNN的分类的时候，准确率一直不是很高，原因就在于df阈值(50)设置的太低，导致词典的长度太短，向量的维度太低，从而使得文档之间的区分度不是很大，因此这次我降低了阈值，重新构建了一个词典，大小为24227。
2. 在测试的时候，我是在训练集和测试集上分开做的，训练集文档数是15056，测试集的文档数为3772。最终结果如下：

Accuracy(train) = 0.8873

Accuracy(train) = 0.9054

* **实验过程中遇到的问题以及解决方法**

在这次动手实践中，由于要计算很多先验知识，用到了许多简单的矩阵除法，而这个看似很小的问题困扰了我很久。由于我使用的是python2.7，矩阵除法是否整除需要自己手动给出定义，对于一个array来说，即使在做除法的过程中进行了数据类型的转化(float)，但存储回原来的array的时候，仍会记作int类型，这就是我在打印出每一个P(term|C)的时候总出现全0的原因，后来和同学讨论之后，经过单步调试，我才明白了其中的道理。这也告诫自己平时一定不要疏于动手，实践和理论是两回事。

* **实验总结**

通过本次实验，我动手实践了贝叶斯的分类过程。动手之前我没有想到会遇到这么多小的麻烦，通过把这个思想用代码的形式展示出来，我发现对于这个分类器有了更深刻的理解。