# 实验报告二

**—————朴素贝叶斯分类器**

201814814 赖文锋

1. 概述

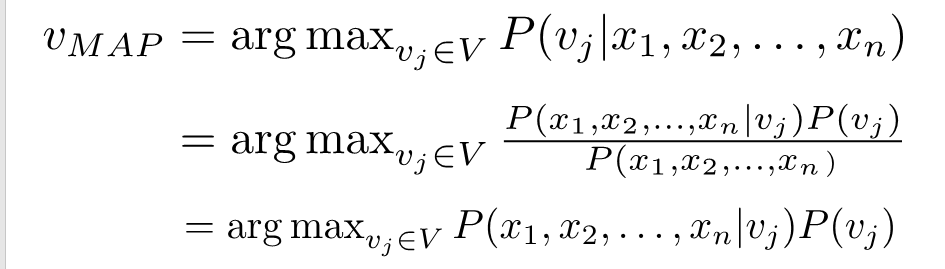
这是实验二的实验报告，主体是一个朴素贝叶斯分类器。尽管很多代码和实验一knn算法很类似，代码上也复用了很大一部分，但其核心依然是完全不同的，朴素贝叶斯分类器主要用的是贝叶斯概率相关理论。同样写在前面的是使用方法：



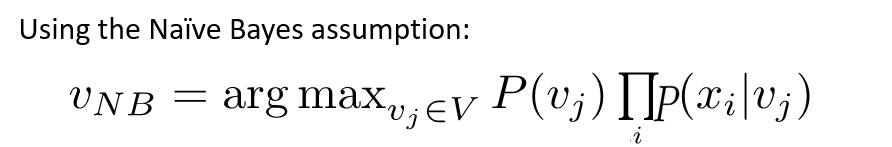
代码的10，11，12行指定了训练集的位置，默认是当前.py文件所在的文件夹的位置，使用者也可以换成自己需要的位置。

1. 主要思想及理论

虽然说核心理论与教科书上的区别并不大，但是依然有一些细节需要阐述。算法的目的是求让P(tag|passage)最大的tag，但是这没办法求，于是我们使用贝叶斯公式转化为P(passage|tag)。



其中最后的联合分布使用一个启发式的公式计算：



那么算法的主要流程也就呼之欲出了。

首先进行预处理环节，这部分是把每篇文章先转化成处理过的单词列表，运用的技术及类库与前一个实验基本相同，得到单词列表再转化为reverse list，也与前一个实验相同。

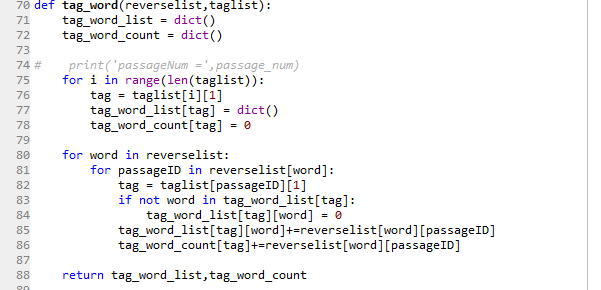
然后开始计算所有的P(词|类标签)，即公式中的P(xi|yj)这部分可以在预测的时候实时计算，但那样会导致每次预测都要重新算一遍，算法时间复杂度剧增，因此在训练过程先算好，存下来。具体计算方式是根据上一步已经得到的reverse list，计算(该词在该类出现的次数/该类的总词数)。接着计算P(类标签)，即公式中的P(vj)，具体计算方法为(有该类标签的文章的数目|文章的总数目)。计算这些数值存下来就已经能进行预测了，但是有一个小问题，如果测试集中出现一个训练集合中没有出现的词，那么会导致P(xi|yj) = (该词在该类出现的次数/该类的总词数)=0，于是我们修改一下定义，意义上相当于默认任何一次词出现的最低概率=(1/训练集总词数)，即P(xi|yj)= (该词在该类出现的次数+1)/(该类的总词数+训练集的总词数)，这种平滑技术可以防止最后算概率的结果=0。

另外考虑不同的词在训练中占的比重不一样，那些在文档中出现的太频繁或者太少见的词，它们重要性可能不那么大，甚至有可能只是分词的一个失败品(许多出现一次的无意义词)，因此考虑用idf给log(P(xi|yj))加权，最终计算公式为P(xi|yj)= (该词在该类出现的次数\*idf+1)/(该类的总词数+训练集的总词数)，其带来的收益在最后结果展示中有分析。

最后是预测阶段，测试文章只需要像训练文章一样进行预处理，然后调用贝叶斯公式计算概率即可，需要用到的结果在之前都已经存

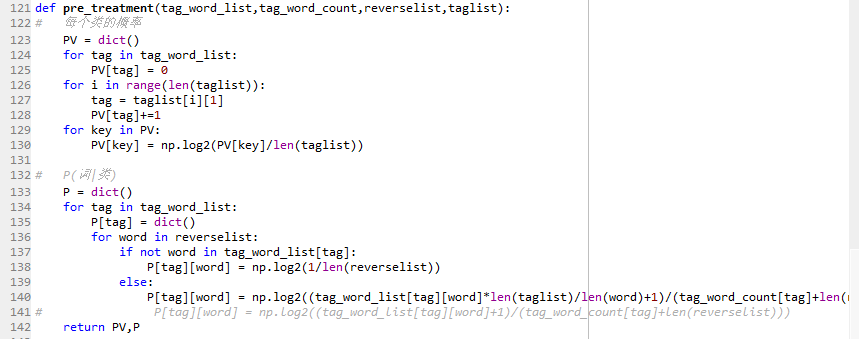
下，最后选择让概率最大的那个标签即可。

1. 主要方法介绍
2. 把文章转化为单词列表与reverse list的方法与上一个实验一致，在此不再赘述。
3. 获得每一个tag里的每个词汇的词频的方法：



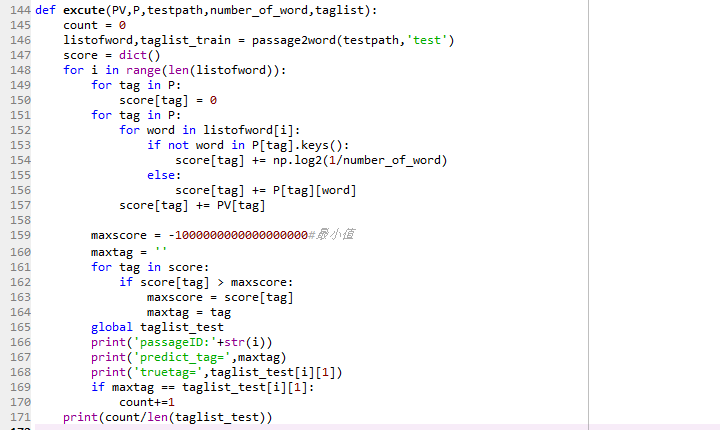
可以看到只是简单的从reverse list把信息重新整合一遍，但reverse list依然有很方便的过滤词频和计算词权重的作用，这些在之后的程序中依然用得上，因此并没有省去这一步。

1. 计算所需的各个数据的方法：



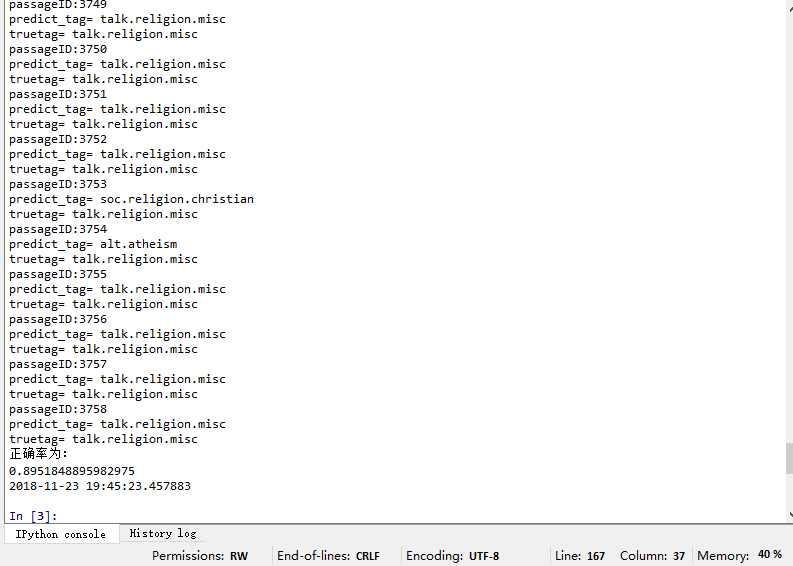
因为在之后的预测中需要用到很长的乘法，因此对于公式取log，把乘变成加是一个很实用的做法，因此在存数据的时候直接存数据取完log的值就可以了。

1. 进行预测的方法：



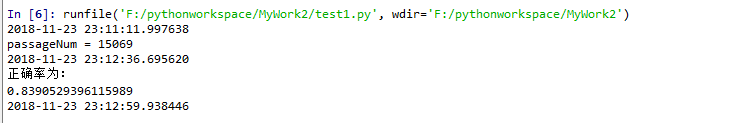
159~168行是计算让概率最高的tag，147~157行是计算每篇文章属于每个tag的概率，用的是贝叶斯公式。最后169~171行计算正确率。

1. 结果展示
2. 预测结果展示：

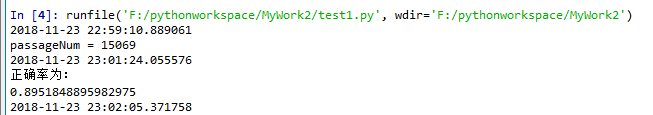


对于每篇文章可以看到预测的tag，真实的tag，最后统计了正确率，正确率接近90%。

1. 一个有趣的对比：如果不给P(xi|yj)加权，即只用词频，结果是这样子的：



如果使用idf对其进行加权，结果如下：



很直观的看到，直接提升了6个百分点！这短短的不到一行的代码，竟然产生如此巨大的改变，足可见idf的重要性(甚至比在knn带来的提升还要明显)。

1. 结果分析与总结。

可以看到该朴素贝叶斯分类器训练的时间大概在60~90s左右(数据量15000)，预测的时间载30s内(数据量3000左右)，应该说运行起来还是挺快的。如果对细节进行优化应该还可以得到快的多的时间。准确率达到了90%，对于这种训练集不算很大的数据应该也算表现不错的。对比knn算法，不仅实现简单，代码量少，而且预处理时间大大缩短，预测时间略有优势，关键是正确率还高了一截，可以说各方面的表现都要优于knn，是个很不错的算法。

在最后感谢指导我的老师和互相交流的同学们，也感谢提出用idf给概率加权的王业超同学，是他给了我灵感让此程序的预测正确率能更进一步。同时通过这次课设也感觉自己对贝叶斯分类器的理解也得到了提升，虽然代码较为简单，但依然很有意义。