作业报告

面向航空安全报告的文档自动分类

一、实验目标

手动实现关于英文文档的多标签分类算法，深入理解不同特征值选取与分类算法对分类结果的影响。

二、实验环境

型号名称：	MacBook Air
型号标识符：	MacBookAir6,2
处理器名称：	Intel Core i5
处理器速度：	1.4 GHz
处理器数目：	1
核总数：	2
L2 缓存（每个核）：	256 KB
L3 缓存：	3 MB
内存：	4 GB
Boot ROM 版本：	MBA61.0099.B22
SMC 版本（系统）：	2.13f15
序列号（系统）：	C02N3B3DG085
硬件 UUID：	E64031A5-9673-5022-86BF-CF22636D6AF4

三、实验要求与数据集信息

（1）数据集

训练集：包括文档数据与分类标记结果。每个文档占txt一行，标记结果以csv显示，属于某类标记为1，否则为-1；

测试集：包括文档数据与分类标记结果，结果用于分析分类结果。

（2）输出

针对测试集，输出每个文档所属类别的标记.

四、实验原理

文本统计分类的流程为：数据预处理 -> 特征计算 -> 特征选择-> 分类学习-> 结果评估。

（1）数据预处理部分，主要目标目标是去杂、整合输入。我在这里引入了英文停用词库，对文档中的单词做了初筛，去除了常见的无意义词，帮助之后提高计算效率；并将csv的信息做了提取，成为更便于阅读理解的txt文档；

（2）特征计算及特征选择部分，主要目标是找出特征词语，改善效果，同时过拟合。我在此采用的是卡方检验法，简要介绍如下：

卡方检验最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。具体做的时候常常先假设两个变量确实是独立的（行话就叫做“原假设”），然后观察实际值（也可以叫做观察值）与理论值（这个理论值是指“如果两者确实独立”的情况下应该有的值）的偏差程度，如果偏差足够小，我们就认为误差是很自然的样本误差，是测量手段不够精确导致或者偶然发生的，两者确确实实是独立的，此时就接受原假设；如果偏差大到一定程度，使得这样的误差不太可能是偶然产生或者测量不精确所致，我们就认为两者实际上是相关的，即否定原假设，而接受备择假设。

举个例子，判断词t与类别c是否相关。t=“篮球”， c=“体育”；统计各类文档数如下：

	体育	¬体育	
包含篮球	A	B	M2
不包含篮球	C	D	
	M1		N

则词t与类别c的卡方值的形式可以写成

$$\chi^2(t, c) = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$$

卡方值越大，说明词t与类别c的相关性越高。因此，我们择优录取就可以得到较为合适的特征项。

针对英文纯文本的实验结果表明：作为特征选择方法时，开方检验和信息增益的效果最佳（相同的分类算法，使用不同的特征选择算法来得到比较结果）；文档频率方法的性能同前两者大体相当，术语强度方法性能一般；互信息方法的性能最差[1]。

（3）分类学习部分，实现了朴素贝叶斯算法，算法原理详见课件不再赘述。

[1] http://www.blogjava.net/zhenandaci/archive/2008/08/31/225966.html

（4）结果分析，采用传统意义上的F值，F值里的准确率指：正确判断出的类别数/判断的总类别数（每个文档可能有好几个）；召回率是指：正确判断出的类别数/测试集正确结果中的总的类别数（每个文档可能有好几个）。

五、实验过程

（1）数据预处理：通过python解析data与result;

（2）特征值计算与分类学习：

特征值计算较为繁琐，之后实现的贝叶斯算法也需统计所有文档，因此可以一次统计多次处理，便于调试，输出中间结果select_meta.txt(词语与类别的文档数统计), select_result.txt（卡方平方的中间结果）, nicewords.txt（排序后的卡方值优秀词语）, basix_table.txt（特征值在各个类别的贝叶斯概率）。

注意，贝叶斯概率计算中，由于概率值小于1，会越乘越小，超出double甚至long double的精度范围，因此，我们在概率上统一成生ENLARG常数（这里设为10^5）。

（3）结果分析：按F值的定义计算F准确率。

六、实验结果与分析

（1）卡方计算后，发现以下词语与22个类别的相关度比较大

gondola(21705) 143.151136, entourage(15905) 143.151136, mit(15386) 143.151136, westport(14828) 143.151136, pointless(8988) 143.151136, revising(14037) 133.772127, meteorology(7958) 107.909669, recite(5031) 94.109560, pumpfailure(8983) 94.109560, industrialengineer(22956) 93.117372, pickup(3707) 91.704383, olu(17107) 87.856931, inoperativeauxiliarypowerunit(8833) 83.518929, lobe(12775) 80.677392, staticairtemperature(13097) 77.727742, buybacks(9837) 74.874384, cerebral(3081) 74.874384, liver(11720) 74.874384, nationalparkservice(14645) 74.874384, hemorrhage(3082) 74.874384, brook(11539) 72.195677, advisor(16926) 72.195677, cumulusbuildup(15295) 72.195677, pupil(3068) 72.005701, poormarking(7307) 72.00570

简单手动参考文本，确实符合直观，这些词语有较为明显的特征性。

（2）朴素贝叶斯结果计算后，所得结果在ans.txt 文档中；F值计算结果在FVALUE.txt文档中。

七、存在问题

可以看到，最终结果的F值并不高，经过反思和小幅调试，我认为主要由如下原因造成：

1、训练与测试的文档长度均有限，维度不高，不很很好支持更多数量的特征值选取；另外，怀疑文档中有些词语分词错误，但情况不普遍，应该无较大影响；

2、实验环境限制，特征值选择偏少，总共有22个类别，特征值在40左右会比较好，现在只能实现了25个。调整特征值数量测试发现，非最高评分的特征值，对各个类别有比较平均的识别性，而这里只能舍弃；

3、许多类别无法识别出来的原因是概率统计出来趋近于零，超出了浮点精度，被直接做零处理，虽然采用了比例放大的方法，但还是效果不好。后查阅资料发现最好使用对数概率

$$x' = \log(x) \in \mathbb{R}$$
$$y' = \log(y) \in \mathbb{R}$$

$$\log(x \cdot y) = \log(x) + \log(y) = x' + y'.$$

然而实践中发现，仍然会有-inf出现；

4、基于以上第三点，我认为自己的特征值选取中仍存在问题。由于是多类别分类，我通过卡方平均选择时，是将所有类别放在一起选择特征值。观察选出的特征值发现，这些特征值均有较强的专一性。但由于是所有类别一起处理，每个类别所选出来的特征词非常少，这应该是造成结果不尽如人意的主要原因。但由于机器跑起来实在太慢，就不便再测试；

设计的优化方案是，对每个类别，选出5-10个特征值，分别判断；

与同学讨论后发现，这种方法下的准确率普遍较现在有明显提高。

七、结论

文本分类是个精细的工作，需要全面考虑噪声处理、特征值提取算法、分类算法等多个问题；各部分算法通力配合才能最终呈现较好的结果。