贝叶斯分类器

设计思路

由于我们需要得到的是给定一片文章来判定他属于哪个分类的问题，即 P(category|doc)
但是目前的根据已知的数据集来看更好知道 P(doc|category) 即在一个分类下，一偏文章的概率。由于每个文章由多个词语构成，每个事件相互独立，我们又可以知道 P(doc|category) = P(word1|category) * P(word2|category) * ….. *P(wordn|category)

由贝叶斯公式得知 P(category|doc) = P(category) * P(doc|category) / P(doc)

P(category) 分类的概率，由于分类中的数据个数相同，随机一个文章属于分类的概率相同 1/20

P(doc) 随机抽取一篇文章的概率也相同，所以问题转换为求 P(doc|category) 概率然后每个分类比大小，最大的即是预测的结果

拆分训练数据和测试数据
取20_newsgroups 中每个分类里前 700 个作为训练数据，后300个作为测试数据

文件分词
分析：每种不同的文件中的文本内容粗略的可以按照空格分开，另外可以过滤或者分开特殊字符类似 ！{} ？，等语气助词和分隔符。这一点在后面测试中发现非常重要，词分的不够好，非常影响测试效果。

通过通用算法 def get_words(doc_path): 读取文件，并按照如下拆分规则进行分解 words = re.split(r'[~`/ ,;{}?!\"\'$#%^&*()\\<>\n\-+\t:_=]', letters)

过滤：
对单词做处理，首先归一化为小写，防止漏判，另外把空字符或者小于1的词语(一般是无意义的 a I 类似的)过滤。另外设置一组 stopwords，把助词语气词都过滤掉

训练数据
遍历所有文件把词语和对应 category 中的数据统计出来：
使用字典 word_category_data 来记录，结构类似 {“love”:{“talk.politics.mideast”: 5, “rec.sport.baseball”: “10”}, “him”:{“talk.politics.mideast”: 1, “rec.autos”:2}}

使用 category_count_data 记录每个分类和分类中包含的所有次的个数

求P(doc|category)，由前面的设计思路知道 P(doc|category) = P(word1|category) * P(word2|category) * ….. *P(wordn|category)

转而求给定一个分类的中每个次的概率
比如 love 这个词在 talk.politics.mideast 的概率， 5/ talk.politics.mideast 中次的总数。需要注意一点的是由于有些次在特定的分类中没有出现，所以我们要对原始概率进行加权，不然会出现0，导致最后结果为0， 所以我们给 5 +1/ talk.politics.mideast + 1 分子分母都加1， 由于分母足够大，对结果没有影响。或者给定一个极小值。

得出给定一个文章中每个次在的概率 p 然后相乘。
由于数据量大，导致一个词在给定的category 中概率太小，乘机会使得数据更小导致后期float超出判断为0，我们取 log 来修正。由于每种结果都是同样的 log 操作，所以对比对大小没有影响

log(p1*p2*p3*p4*...pn) = log(p1) + log(p2) + log(p3) ….+log(pn)

从而得到每个文章的相对概率

测试数据
拿剩下的30% 数据进行测试。算出特定分类下，判断正确的个数 / 总个数，打出每个分类判断的概率。

代码实现

Online Code:
https://github.com/jackrex/AILesson/tree/master/L2

测试结果

增加一定优化

sci.crypt prob is： 0.97
comp.sys.mac.hardware prob is： 0.416666666667
talk.politics.misc prob is： 0.956666666667
soc.religion.christian prob is： 0.993265993266
rec.motorcycles prob is： 0.783333333333
sci.med prob is： 0.46
comp.graphics prob is： 0.793333333333
comp.windows.x prob is： 0.81
comp.sys.ibm.pc.hardware prob is： 0.48
talk.politics.guns prob is： 0.73
alt.atheism prob is： 0.456666666667

comp.os.ms-windows.misc prob is： 0.94
sci.space prob is： 0.71
talk.religion.misc prob is： 0.64
misc.forsale prob is： 0.59
rec.sport.hockey prob is： 0.816666666667
rec.sport.baseball prob is： 0.413333333333
talk.politics.mideast prob is： 0.973333333333
rec.autos prob is： 0.323333333333
sci.electronics prob is： 0.703333333333

记录和总结（第一次测试）

1. 在使用了 stopwords 过滤词之后，rec.autos 、comp.sys.mac.hardware、rec.motorcycles 、 sci.electronics 等有一定幅度概率提升，猜测这些分类中语气助词较多，受到干扰颇多

使用前
talk.politics.mideast prob is： 0.88
rec.autos prob is： 0.07
comp.sys.mac.hardware prob is： 0.13
alt.atheism prob is： 0.36
rec.sport.baseball prob is： 0.18
comp.os.ms-windows.misc prob is： 0.05
rec.sport.hockey prob is： 0.5
sci.crypt prob is： 0.56
sci.med prob is： 0.24
talk.politics.misc prob is： 0.73
rec.motorcycles prob is： 0.07
comp.windows.x prob is： 0.71
comp.graphics prob is： 0.52
comp.sys.ibm.pc.hardware prob is： 0.37
sci.electronics prob is： 0.08
talk.politics.guns prob is： 0.19
sci.space prob is： 0.29
soc.religion.christian prob is： 0.72
misc.forsale prob is： 0.2
talk.religion.misc prob is： 0.1

使用后
talk.politics.mideast prob is： 0.83

rec.autos prob is : 0.47
comp.sys.mac.hardware prob is : 0.61
alt.atheism prob is : 0.63
rec.sport.baseball prob is : 0.69
comp.os.ms-windows.misc prob is : 0.38
rec.sport.hockey prob is : 0.81
sci.crypt prob is : 0.74
sci.med prob is : 0.69
talk.politics.misc prob is : 0.77
rec.motorcycles prob is : 0.54
comp.windows.x prob is : 0.87
comp.graphics prob is : 0.66
comp.sys.ibm.pc.hardware prob is : 0.7
sci.electronics prob is : 0.47
talk.politics.guns prob is : 0.54
sci.space prob is : 0.72
soc.religion.christian prob is : 0.71
misc.forsale prob is : 0.45
talk.religion.misc prob is : 0.23

2. 删除letter 最后一行作者签名相关的部分的影响
测试结果影响并不大，有些部分有较小提升或减弱

3. 字母转大小写对判断概率的影响
有提升，提升范围大概是 1% - 8% 左右

4. 提高训练数据多少，由all data 70% 提升到 80% - 90% 测试由 30% 减少到 20% - 10%
大部分提升，小部分下降 范围不超过 前后 8%

5. 增加了包含数字过滤的word 过滤
提升4% 的点