

# Data Mining 实验报告

## Homework 2 : 朴素贝叶斯分类器的实现

学号：201844906

姓名：宋春娇

### 一、实验要求

按照 8:2 的比例对 20 个新闻组数据集划分为训练集和测试集（要求测试数据均匀覆盖各个类别），应用朴素贝叶斯分类器对测试数据进行分类，统计分类准确率。

### 二、朴素贝叶斯分类器

1. 基本思想：对于待分类项，求解在该项出现的条件下各个类别出现的概率，哪个概率最大，就认为该分类项属于哪个类别。

2. 正式定义

1、设  $x = \{a_1, a_2, \dots, a_m\}$  为一个待分类项，而每个  $a$  为  $x$  的一个特征属性。

2、有类别集合  $C = \{y_1, y_2, \dots, y_n\}$ 。

3、计算  $P(y_1|x), P(y_2|x), \dots, P(y_n|x)$ 。

4、如果  $P(y_k|x) = \max\{P(y_1|x), P(y_2|x), \dots, P(y_n|x)\}$ ，则  $x \in y_k$ 。

关键就是如何计算 3 中的各个条件概率。由条件概率可知：

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

忽略分母，分子中的  $P(x|y_i)$  转换为某文档中每个词在各个类别中出现概

率的乘积。即分子=

$$P(y_i) \prod_{j=1}^m P(a_j|y_i)$$

### 3. 平滑技术

$$P(\text{"发票"}|S) = \frac{\text{每封垃圾邮件中出现“发票”的次数的总和}+1}{\text{每封垃圾邮件中所有词出现次数(计算重复次数)的总和}+\text{被统计的词典的词语数量}}$$

多项式平滑技术 :  $P(\text{词 } w|\text{类 } A) = (\text{词 } w \text{ 在类 } A \text{ 中出现的总次数}+1) / (\text{类 } A \text{ 中所有词出现的次数}+\text{类 } A \text{ 词典数})$

### 4. 具体实现

#### (1) 为训练集建立向量

将训练集内每一个类别表示成一个向量, 格式为[类名, 类中单词总数, 类出现的概率, 类的单词字典], 训练集为一个 list, 每个元素是各个类别的向量。

①由于每个类别出现的概率= (该类中的文档总数) / (训练集中文档总数), 所以编写函数 count\_files (), 用来计算训练集内文档总数。运行程序可知: 训练集文档总数: 15056。

②遍历训练集内的每一个文件夹(即类别)时, 需要统计类中单词总数, 这需要遍历每个文档时进行累加; 类的词典应包括该类中每个文档中出现的单词以及它们出现的总次数。值得注意的是类中单词总数以及类的字典要在遍历完该类中所有文件之后再追加进该类的列表中。运行该函数后生成的部分训练集向量如图 1 所示。

```
第15个向量:
['sci.space', 145231, 0.052404357066950055, {'yellow': 5, 'interchang': 1, 'four': 48, 'prefix': 16, 'tr
第16个向量:
['soc.religion.christian', 161991, 0.052935706695005316, {'parenthetc': 1, 'osiri': 11, 'foul': 9, 'inte
第17个向量:
['talk.politics.guns', 148867, 0.048352816153028694, {'children': 1, 'yellow': 3, 'four': 28, 'authorit'
第18个向量:
['talk.politics.mideast', 203207, 0.04994686503719448, {'fawn': 2, 'zurueckfuehren': 1, 'convic': 3, 'fo
第19个向量: |
['talk.politics.misc', 150318, 0.04117959617428268, {'orthogon': 2, 'clerali': 1, 'woodi': 9, 'osiri': 1
第20个向量:
['talk.religion.misc', 95065, 0.033342189160467585, {'moskowitz': 2, 'osiri': 4, 'yellow': 2, 'four': 20
```

图 1 创建的部分训练集向量

## (2) 为训练集建立向量

将测试集的每一个文档表示成向量, [类名, 以该文档所有单词为元素的列表]。运行该函数后生成的部分测试集向量如图 2 所示。

```
第3770个向量:
['talk.religion.misc', ['bill', 'world', 'would', 'better', 'connect', 'pharvey', 'propheci', 'form',
第3771个向量:
['talk.religion.misc', ['among', 'individu', 'true', 'massada', 'cunyv', 'gerri', 'common', 'truth',
第3772个向量:
['talk.religion.misc', ['bill', 'someth', 'mif', 'tek', 'gag', 'bil', 'kfu', 'reign', 'funni', 'thi',
```

图 2 创建的部分测试集向量

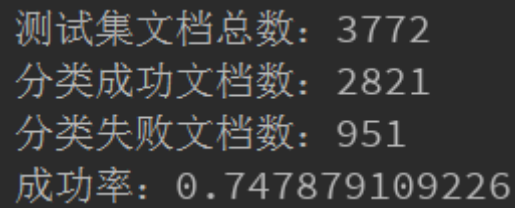
## (3) 为训练集建立向量

实现 NBC 的主要操作是计算后验概率  $P\{c_i|x\}$ (其中  $C_i$  指某一类别,  $x$  指待分类的文档)。该后验概率的计算已转换为计算乘积:  $P\{C_i\}$ \*每个词在该类别中出现的概率。

为了避免乘积结果超出计算机计算的下限, 采用取对数操作, 将乘法转换为加法。

对于每一个文档, 要计算该文档出现时各个类别出现的概率, 所以每个文档有一个列表, 列表元素为元组, 元组元素分别为类别名和概率。使用 sort 函数对列表根据概率值进行排序, 排序后列表第一个元组中的类名即朴素贝叶斯算法对该文档的分类结果。统计所有文档的成功次数与失败次数, 计算准确率。

运行该函数截图如图 3 所示。



```
测试集文档总数：3772
分类成功文档数：2821
分类失败文档数：951
成功率：0.747879109226
```

图3 N B C 算法性能

### 三、实验总结

朴素贝叶斯算法与其他机器学习中的算法不同，像如 KNN 算法等，均是直接学习出特征输出  $Y$  和特征  $X$  之间的关系；但是朴素贝叶斯是直接找出特征输出  $Y$  和特征  $X$  的联合分布  $P(X,Y)$ ，然后用  $P(Y|X)=P(X,Y)/P(X)$  得出。朴素贝叶斯思想简单直观，计算量也不大，在文本分类中具有不错的分类效果。

值得注意的是，朴素贝叶斯模型给定输出类别的情况下，假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。