山东大学

信息科学与工程学院

2022－2023 学年第一学期

实 验 报 告

课程名称：信息基础 II

专 业 班 级 崇新学堂

学 生 学 号

学 生 姓 名

课 程 报 告 朴素贝叶斯

1. 实验目的

《联邦党人文集》中总计有 85 篇文章，其中有 12 篇作者未知，一共有三个撰写人：*JAY/HAMILTON/MADISON*,模型上采用朴素贝叶斯，手工提取文本特征，使用已知作者文章作为训练集对模型进行训练，之后对未知作者的 12 篇文章进行作者分类。

2. 实验要求

不能直接使用 ML 库或开源代码

3. 实验原理

3.1 朴素贝叶斯

所谓朴素贝叶斯，可以拆分开理解，“朴素”代表特征之间相互独立互不影响，在本次实验中体现在不同单词出现与否是相互独立的，由此根据变量相互独立的性质，可以得到公式（1）.

$$P(x_1, x_2 \cdots x_M \mid y) = P(x_1 \mid y) \cdot P(x_2 \mid y) \cdot \ldots \cdot P(x_M \mid y) \tag{1}$$

“贝叶斯”代表着贝叶斯公式，如公式（2）

$$P(c \mid w) = \frac{P(w \mid c) \cdot P(c)}{P(w)} \tag{2}$$

从而可以得出朴素贝叶斯的公式：如公式（3）

$$P(y \mid x_1, x_2 \cdots x_M) = \frac{P(x_1 \mid y) \cdot P(x_2 \mid y) \cdot \ldots \cdot P(x_M \mid y) \cdot P(y)}{P(x_1, x_2 \cdots x_M)} \tag{3}$$

3.2 拉普拉斯平滑（ADD-1）

在 3.1 节的讨论中，确定了朴素贝叶斯的基本思路，但在数据集数量不够大的情况下，容易出现一些特殊情况，如 $P(x_i \mid y) = 0$，出现这种情况之后，所有的概率相乘之后就会为 0，无法进行预测，此时采用拉普拉斯平滑来解决这类问题，如公式（4）.其中 α 为拉普拉斯平滑系数，M 为类别总数。

$$P(x_1 \mid y) = \frac{N_{x_1, y} + \alpha}{N_y + \alpha M} \tag{4}$$

4. 实验过程

4.1 特征提取

针对本次实验内容，共计 85 篇文章，其中存疑对象只有 *HAMILTON*、*MADISON*，因此在处理数据集时，剔除 JAY 的文章，而只将 *HAMILTON*、*MADISON* 写的 66 篇文章作为数据集，未知作者的 11 篇文章（老师说有 12 篇，但我好像只找到了 11 篇）作为预测对象。随后对文章进行特征提取。

4.1.1 特征词选取

在本次实验中，特征词的选取是数据处理的关键环节，我在本次实验中阅读了相关论文(Jockers & Witten, 2010)，确定了我的关键词。关键词列表如下：

```
features = ['a','america','an','and','arms','be','more','been','by','confederacies','four'
            'given','has','importance','in','is','independent','nations','of','on','one','only'
            ,'others','powers','soon','that','the','them','there','they','this','three',
            'to','treaties','upon','useful','well','which','wise']
```

4.1.2 获取特征向量

在确定了特征词之后，则需要读取数据，判断对应文章中是否存在特征词中的单词，如果存在，则对应位置为1，反之则为0：实现代码如下：

```
'''特征提取'''
alphabet = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
features = ['a','america','an','and','arms','be','more','been','by','confederacies','four'
            'given','has','importance','in','is','independent','nations','of','on','one','only'
            ,'others','powers','soon','that','the','them','there','they','this','three',
            'to','treaties','upon','useful','well','which','wise']
Fea={}
for i in range(len(features)):
    Fea[features[i]]=0
paper='''<此处为《联邦党人文集》内容>'''
word=paper.lower()                          #将文章中的字母全部变为小写
for i in range(len(word)):
    if word[i] not in alphabet:
        word=word[:i]+' '+word[i+1:]        #将非字母的单位如标点符号等转化为空格
word=word.split()                           #利用split函数将字符串拆分为单词
for i in range(len(word)):                  #随后对特征词进行验证，文章中是否存在，并且计数
    for j in Fea.keys():
        if word[i]==j:
            Fea[j]=Fea[j]+1
print(list(Fea.values()))                   #获得了特征向量
```

以上代码的主要思路是将每一篇的文章先全部转化为小写，之后将标点符号等转化为空格，之后利用split（）函数将文章拆分为单词列表，从而进行特征词的判断和计数，（这部分代码其实完全可以使用 in 判断即可，但是我当时为了能够实现计数，写了这部分代码，在实际后面的朴素贝叶斯中，其实不用知道数量，但这部分代码我还是保留了下来）。利用该代码便获得了特征向量。

4.2 朴素贝叶斯

4.2.1 训练过程

1. 求取$P(y)$是一个相对比较简单的过程，只需求取样本数量，之后根据 label 将训练集分为两类，用对应分类下的样本数量除以总的样本数量即可获得$P(y)$，代码中：one 函数代表将之前获得的计数版本的特征向量中全部非 0 数字变为 1。

```
#----------------------------训练过程-----------------------------------------------
one(data,test)                                  #将特征向量归一
sample_number = len(data)                       #训练集数量
feature_number = len(data[1])                   #特征词数量
HAMILTON_number=class_probability()             #训练集中作者为HAMILTON的样本数量
P_HAMILTON =HAMILTON_number/len(target)         #HAMILTON为作者的概率
P_MADISON = 1-P_HAMILTON                        #MADISON为作者的概率
#----------------------------获取分类索引-------------------------------------------
```

2.分类求取条件概率：第二步的主要任务是求取$P(x|y)$,具体思路为：首先是获得每个类别下的累计特征向量，即将每个类别下的特征向量相加，从而获得了出现次数向量，这里便获得了条件概率的分子（对分子加 1 是拉普拉斯平滑的结果），之后除以每个类别的总数，即可获得条件概率向量。

```
#----------------------------获取分类索引-------------------------------------
index0_set = []
index1_set = []
for i in range(len(target)):                #获取不同类别的数据索引
    if target[i] == 0:
        index0_set.append(i)
    else:
        index1_set.append(i)
#----------------------------分类求取条件概率---------------------------------
p0_num = [1 for _ in range(feature_number)] #第120，121，128，129行代码进行的过程是拉普拉斯平滑过程
p1_num = [1 for _ in range(feature_number)]
for i in index0_set:
    for j in range(feature_number):
        p0_num[j] =p0_num[j] + data[i][j]   #求取概率分子，及对应单词出现的总次数
for i in index1_set:
    for j in range(feature_number):
        p1_num[j] =p1_num[j] + data[i][j]
sum0=0
sum1=0
for i in index0_set:
    for j in range(feature_number):
        sum0+=data[i][j]
for i in index1_set:
    for j in range(feature_number):
        sum1+=data[i][j]
p0_den = 2+sum0     #拉普拉斯平滑因子取1，平滑项使用类别总数
p1_den = 2+sum1
p0 = [1 for _ in range(feature_number)]
p1 = [1 for _ in range(feature_number)]
for i in range(len(p0_num)):
    p0[i]=math.log(p0_num[i]/p0_den)
for i in range(len(p1_num)):
    p1[i]=math.log(p1_num[i]/p1_den)
```

在此处拉普拉斯平滑系数取 1，同时在判断的过程中，若干个很小的概率相乘在一起会变得非常小而近似于 0，由于最后的判决只需要判断大小即可，无需求出具体的数值，所以在此处我们对概率求对数，之后的相乘也就变成了相加从而能够解决概率过小乘积趋近于 0 的问题。

4.2.2 检验过程

利用训练集进行检验，获得的正确率为：80.3%，相对比较理想。

```
y = [1 for _ in range(len(data))]
pri0 = [math.log(P_HAMILTON) for _ in range(len(data))]
pri1 = [math.log(P_MADISON) for _ in range(len(data))]
for i in range(len(data)):
    for j in range(feature_number):
        if data[i][j]==1:
            pri1[i]=p1[j]+pri1[i]
            pri0[i]=p0[j]+pri0[i]
for i in range(len(data)):
    if pri0[i]>pri1[i]:
        y[i]=0
    else:
        y[i]=1
n=0
for i in range(len(data)):

    if y[i]==target[i]:
        n+=1
print('正确率为：',n/len(data))
```

```
正确率为： 0.803030303030303
```

4.2.3 预测过程

之后将之前处理好的预测数据的特征向量引入模型进行预测，预测代码如下：预测结果如表 1 所示。

```
#------------------------------预测-----------------------------------------------------
label = [1 for _ in range(len(test))]
pr0 = [math.log(P_HAMILTON) for _ in range(len(test))]
pr1 = [math.log(P_MADISON) for _ in range(len(test))]
for j in range(feature_number):
    for i in range(len(test)):
        if test[i][j]==1:
            pr1[i]=p1[j]+pr1[i]
            pr0[i]=p0[j]+pr0[i]
for i in range(len(test)):
    if pr0[i]>pr1[i]:
        label[i]=0
    else:
        label[i]=1
print(label)
```

```
[1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

表 1 预测结果

No.49	*MADISON*
No.50	*HAMILTON*
No.51	*MADISON*
No.52	*MADISON*
No.53	*MADISON*
No.54	*MADISON*
No.55	*MADISON*
No.56	*MADISON*
No.57	*MADISON*
No.62	*MADISON*
No.63	*MADISON*

可以发现预测结果与论文中基本一致，但是仍有一篇文章被分类为 *HAMILTON*，与原论文不符，初步分析原因为选择的特征词数量过多，造成了过拟合现象，从而出现了错误。

5. 实验心得

在编程实现的过程中，我发现有很多的现有库函数，包括文本特征提取函数，但是要求中不可以使用，于是便查询了较多论文，确定了特征词，之后自己编写代码进行特征提取，通过编程实现的整个过程，我更好地理解了老师的公式推导，虽然整体代码难度都不大，其中有很多的先验知识的支撑，包括拉普拉斯系数的选择，通过自己编程也更好地掌握了朴素贝叶斯的相关知识及整个机器学习的过程，收获良多。辛苦学姐!

References:

Jockers, M. L., & Witten, D. M. (2010). A comparative study of machine learning methods for authorship attribution. *LITERARY AND LINGUISTIC COMPUTING*, 25(2), 215-223. http://doi.org/10.1093/llc/fqq001