# Bayes for Spam Classification

2017011479 胡锦毅

## 一、模型概述

使用朴素贝叶斯算法，实现一个垃圾邮件分类系统，用于对邮件进行分类。

## 二、算法原理及设计方法

### 1. 算法原理

基于属性条件的独立性假设，我们可以得出：

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

本系统中，我们把$x_i$看作邮件中的词，$y$为spam或者ham。于是，我们的训练目标即为找到：

$$\hat{y} = \arg\max_{\theta} P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

基于以上原理和极大似然估计的思想，我们从训练集中的数据对$P(x_i \mid y)$进行估计。

### 2. 算法流程

- 对分词后的文件进行处理，过滤掉标点符号以及其他无关字符，同时去掉了只有一个字的词，方便处理
- 对每篇文章所含有的词建一个list，作为每篇文章的word vector，这部分没有去重，保留了词频
- 将所有文章的word vector进行整合，建立了vocabulary。其中每个词记为$s_i$
- 训练过程：对vocabulary中每个词在垃圾邮件和非垃圾邮件出现的次数进行统计，并分别求出这个词出现在垃圾邮件和非垃圾邮件的条件概率，计算方法如下

$$p(s_i \mid spam) = \frac{times\ of\ occurence\ of\ s_i + 1}{times\ of\ occurence\ of\ s_i + total\ number\ of\ spam\ emails + 2}$$

$$p(s_i \mid ham) = \frac{times\ of\ occurence\ of\ s_i + 1}{times\ of\ occurence\ of\ s_i + total\ number\ of\ ham\ emails + 2}$$

- 测试过程：对待检验的邮件所有词的概率取log后相加，再分别将垃圾邮件所占比例和非垃圾邮件所占比例的对数，得到两个似然函数

的结果。若垃圾邮件似然较大，则判定为垃圾邮件，反之则判定为非垃圾邮件。

3. 训练方法：

本程序采用了五折交叉验证，即将数据集随机的平均分成五份，将其中对四份作为训练集，另外一份数据集作为测试集，分别训练并测试五次，算出五次的平均值作为最后的结果。

具体运行方法见README.md

## 三、模型分析

在模型的实现过程中，主要进行了两次改进，两次改进都是对$p(s_i|y)$估计方法进行的修改，且都是围绕词频对于概率产生的影响进行的改变

- 修改一：

  在最初的实现版本中，$p(s_i|y)$的计算方法为：

$$p(s_i|ham) = \frac{times\ of\ occurence\ of\ s_i + 1}{total\ number\ of\ ham\ emails + 2}$$

$$p(s_i|spam) = \frac{times\ of\ occurence\ of\ s_i + 1}{total\ number\ of\ spam\ emails + 2}$$

这种方法虽然争取率并不低，可以达到97.3%，但是仔细分析来看，这种概率计算的方法并不符合道理。

于是，在暂时没有想出更好的解决方案之前，决定暂时采用不考虑频率的方法，即一个词在一封邮件只要出现就记为1，未出现就记为0，这样概率中用邮件的个数遍符合道理。

- 修改二：

上述方法虽然符合道理，但是并没有考虑词在一封邮件中出现的频率，准确率自然会下降。于是我进行了第二次修改，尝试将频率作为权重加入到概率的计算中，于是将概率改进为上述概率的计算方法，效果有明显提升

## 四、测试结果

### 1. 评价标准

除了使用准确率作为评价标准外，程序中还使用了精确率(precision)，recall(召回率)，F1分数作为指标进行评测。

## 2. 测试结果

五折交叉验证结果：

| 准确率 | 精确率 | 召回率 | F1分数 |
|---|---|---|---|
| 97.53% | 97.86% | 98.45% | 98.15% |

部分训练集测试结果

| 训练集比例 | 平均值 | 最小值 | 最大值 |
|---|---|---|---|
| 1 | 97.83% | 97.83% | 97.83% |
| 0.8 | 97.75% | 97.72% | 97.78% |
| 0.5 | 97.58% | 97.53% | 97.65% |
| 0.05 | 96.71% | 96.51% | 96.92% |

# 五、Issue

## 1.训练集大小对实验的影响

| 训练集比例 | 平均值 | 最小值 | 最大值 |
|---|---|---|---|
| 1 | 97.83% | 97.83% | 97.83% |
| 0.8 | 97.75% | 97.72% | 97.78% |
| 0.5 | 97.58% | 97.53% | 97.65% |
| 0.05 | 96.71% | 96.51% | 96.92% |

从上表可以看出，准确率随着训练集大小的增大而上升，且越接近1上升速度越缓慢。从总体上来看，数据集对正确率的影响并不多，即使在只有0.05训练集的情况下，仍然有近96.7%的正确率。可见只要数据集本身数量较大，便可以达到一定的效果。贝叶斯方法具有较强的鲁棒性。

## 2.平滑性的解决方法

对于未在字典中出现的词，程序中采用了Laplace平滑的方法来计算概率，计算方法如下：

对于本身概率为：

$$P = \frac{a}{b}$$

修改为：

$$P = \frac{a+1}{b+2}$$

这样，当a=0时也不会出现概率为零的情况

### 3. 特殊特征：邮箱地址

程序对邮件发送地址的第一部分进行了提取，即对于发送地址为xxx@163.com，只提取@163这一部分。将这一部分加入到分词中共同作为特征, 正确性有所提升。

具体结果如下：

| 是否增加特征 | 准确率 | 精确率 | 召回率 | F1分数 |
| --- | --- | --- | --- | --- |
| 增加特征 | 97.53% | 97.86% | 98.45% | 98.15% |
| 不增加特征 | 97.23% | 97.48% | 98.38% | 97.93% |

## 六、Honor Code

本次作业在算法思路层面与计74班陈果(2017012177)和计74班梁念宁(2017011417)两位同学进行过讨论。