**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： CS1803

学 号： U201814792

姓 名： 宋家垚

成 绩：

指导教师： 黄宏

**完成日期： 2020年 6月 18日**

# 实验二

## 实验题目：Naïve Bayes（朴素贝叶斯）

## 实验要求

动手实现Naïve Bayes算法，本次实验使用一个较小的英文数据集（50个实例）和中文数据集“The TREC 2006 Chinese Public Corpus – 60MB(trec06c.tgz)”其中有60000+多个邮件，垃圾邮件和正常邮件的比例为2：1。由于因为编码原因有许多邮件可能采用了混合编码（非gb2312），所以一些邮件打开之后出现非常生僻的中文字符。在本次实验中可以直接忽略这些文件，所以剩下的有用文件一共有51069个。这也是总的数据集大小。

数据处理要求：

* 要求对英文你邮件进行简单停用词处理。
* 要求对中文邮件进行分词处理，简单的停用词处理。

参数估计（至少选一种）：

* 离散特征
* 多项式特征
* 连续特征

最终结果汇总在一个文档中即可。

## 算法设计

Naïve Bayes的算法思想：假设一封新邮件是垃圾邮件和不是垃圾邮件的概率为50%，即先验概率为50%，然后根据条件概率公式，计算当单词W已经存在的条件下，该邮件为垃圾邮件的概率有多大。选出一封邮件中影响成为垃圾邮件概率最大的15个单词，计算它们的联合概率，若有的词是第一次出现，则假定该概率为0.4。根据最终的联合概率判断该邮件是否为垃圾邮件，若概率小于0.9，就表示是正常邮件，否则为垃圾邮件。

本次实验需要处理两个数据集，一个是英文邮件数据集，一个是中文邮件数据集，我选择将两个数据集分开处理，英文邮件集的停用词处理使用了nltk自然语言处理第三方库，测试集采用了每次抽取10个测试中文邮件的停用词处理和分词处理使用了jieba第三方库。

## 实验环境与平台

操作系统：MacOS Catalina 10.15.5

编辑器：VSCode 1.45.1

Python版本：3.7.7

## 程序实现

（英文邮件处理部分）

* 1. 构造测试集，将训练集打乱后取出前十个作为该次测试集，本次测试结束后将测试集文件夹内的文件复位，并进行下一次测试，共进行100次测试，测试完成后进入步骤5。
  2. 获取训练集邮件的单词列表以及单词集，并做降噪处理，最终得到垃圾邮件单词集，正常邮件单词集，总单词集，共3个集合。
  3. 分别计算总单词集中的单词在ham中的出现频率以及在spam中的出现频率，若对应分类单词集中没有该单词则默认将其频率设置为0.01。最终得到两个单词集对应频率的字典。
  4. 按照朴素贝叶斯的算法思想，对测试集内的邮件进行分类判断，并计算准确率，分类完成后记录准确率，回到步骤1。
  5. 画出100次判别的准确率散点图，观察整体准确率情况。

（中文邮件处理部分）

1. 分别获得正常邮件、垃圾邮件以及测试文件的名称列表。
2. 获取训练集中正常邮件与垃圾邮件的数量。
3. 获取停用词表，用于对停用词过滤。
4. 分别获取正常邮件以及垃圾邮件中的词频。
5. 对测试集中的邮件进行测试并计算准确率。

## 实验结果

电脑屏幕的截图

描述已自动生成

图1 英文邮件分类准确率散点图

手机屏幕的截图

描述已自动生成

图2 中文邮件分类准确率结果

## 结果分析

算法正确实现的情况下，整个实验的难度其实主要是数据集的预处理部分，如何将数据集整理成所需要的形式是第一大问题，朴素贝叶斯的算法的实现其实相对而言就比较简单，也比较好理解。

在英文邮件的处理部分，由于数据集比较小，所以采用了小测试集，多次测试的方法来尽量随机的检验准确度，邮件的自然语言处理部分也比较容易，只需要将停用词去除即可，当然，在实验过程中发现，去停用词和不去停用词的准确率差距并不大，甚至可以忽略，相比起来，对于单词集中没有的单词的默认概率对结果的影响比较大，当其为0.01是效果最佳。

在中文邮件的处理部分，数据集大，文件散乱，给文件的预处理带来了额外的难度，需要将数据集分好类才能进行相关的学习和测试。其次，由于中文的特殊性，需要对邮件进行分词处理，这里调用了第三方库来进行辅助，也消耗了一定的学习成本。在分类算法部分和英文邮件大同小异，相对而言不是那么困难。

# 参考文献

[1]Galoa. 基于朴素贝叶斯+Python实现垃圾邮件分类和结果分析[EB/OL]. https://blog.csdn.net/Galoa/article/details/79871992

[2]赵大寳. 朴素贝叶斯实战中文垃圾邮件分类[EB/OL]. https://fuhailin.github.io/NaiveBayes-Spam-Classification/