# Lab6 NaiveBayes 应用实践

**实验目的：**使用概率分布进行分类；实现朴素贝叶斯分类器及其应用。

**实验简介：**检测垃圾邮件的问题等价于，判断每个邮件是垃圾邮件的可能性大还是非垃圾邮件的可能性大。利用 python 的文本处理能力将文档切分成词向量，然后利用词向量对文档进行分类。代码文件：bayes.py

## 1. 准备数据：从文本中构建词向量

在 email/spam 文件夹中有 25 封垃圾邮件，在 email/ham 中有 25 封正常邮件，将其进行垃圾邮件分类。

### （1）分词：切分文本

首先遇到的问题是怎样把一封邮件进行分词，即将其划分成一个个单词形式。函数 textParse()实现将一个长的字符串进行分词的操作。

#对于文本字符串，可用 string.split()方法将其切分：

<<< mySent='This book is the best book on Python or M.L. I have ever laid eyes upon.'

<<< mySent.split()

#可用正则表达式切分，其中的分隔符是除单词、数字外的任意字符串(关于正则表达式可以参考网上的资)。

<<< import re

<<< regEx=re.compile('\\w*')

<<< listOfTokens= regEx.split(mySent)

<<< listOfTokens

<<< [tok in listOfTokens if len(tok)>0]

<<< [tok.lower() for tok in listOfTokens if len(tok)>0]

<<< emailText=open('email/ham/6.txt').read()

<<< listOfTokens= regEx.split(emailText)

### （2）生成词汇表

将所有的邮件分词后生成一个 dataSet，然后生成一个词汇表，这个词汇表是一个集合，即每个单词只出现一次，词汇表是一个列表形式如：

["cute","love",help",garbage","quit"…]。



WangBianqin, Public Laboratory & Teaching Center, Guangzhou East Campus, Sun Yat-sen University

# 函数 createVocabList 运行效果：

```
<<< import bayes
<<< listOPost, listClasses=loadDataSet()
<<< myVocabList=bayes.createVocabList(listOPost)
<<< myVocabList
```

**（3）生成词向量**

每一封邮件的词汇都存在了词汇表中，因此可以将每一封邮件生成一个词向量，存在几个则为几，不存在为 0，例如：[“love”,”garbage”]，则它的词向量为[0,1,0,1,0,…]，其位置是与词汇表所对应的，因此词向量的维度与词汇表相同。

# 函数 bagOfWords2Vec()运行效果：

```
<<< bayes.bagOfWords2Vec(myVocabList, listOPost[0])
<<< bayes.bagOfWords2Vec(myVocabList, listOPost[3])
```

## 2. 训练算法-从词向量计算概率

训练模型：在训练样本中计算先验概率 p(Ci)和条件概率 p(x,y | Ci)，本实例有 0 和 1 两个类别，所以返回 p(x,y | 0)，p(x,y | 1)和 p(Ci)：

（1）若有的类别没有出现，其概率为 0，会十分影响分类器的性能。所以采取各类别默认 1 次累加，总类别（两类）次数 2，这样不影响相对大小。

（2）若很小是数字相乘，则结果会更小，再四舍五入存在误差，而且会造成下溢出。采用取 log，乘法变为加法，并且相对大小趋势不变。

#函数 train()完成训练

```
<<< from numpy import *
<<< reload(bayes)
<<< listOPost,listClasses=bayes.loadDataSet()

<<< trainMat[]
<<< for postinDoc in listOPost:
        trainMat.append(bayes.bagOfWords2Vec (myVocabList,postinDoc))
<<< p0V,p1V,pAb=train(trainMat,listClasses)
<<< pAb
<<< p0V
<<< p1V
```

## 3. 测试过程-根据现实情况修改分类器

首先将 50 封邮件（25 封正常邮件和 25 封垃圾邮件）读进 docList 列表中，然后生成一个词汇表包含所有的单词，接下来使用交叉验证，随机的选择 10 个样本进行测试，40 个样本进行训练。

训练模型：40 封训练样本，训练出先验概率和条件概率；测试模型：遍历 10 个测试样本，计算垃圾邮件分类的正确率。

#函数 spamTest()完成测试。

```
<<< reload(bayes)
<<< bayes.testingNB()
<<< bayes.spamTest()
<<< bayes.spamTest()
```

由于随机选择样本，可以运行 10 次取平均值。注意，这里一直出现的是将垃圾邮件误判为正常邮件（False Positive），这会比将正常的误判为垃圾邮件（False Negative）要好。

## 4. 完成习题

（1）NB 算法的基本思想是什么？

（2）实验中如何解决零概率问题？

（3）如何解决概率值太小会产生溢出问题？