



本科生毕业论文（设计）

基于朴素贝叶斯算法的垃圾邮件过滤

**Spam Filtering Based on Naive Bayesian
Algorithm**

专 业_____电子信息工程_____

姓 名_____罗鹏娟_____

学 号_____15033114_____

指 导 教 师_____王 博_____

完 成 时 间_____2019 年 6 月_____

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包括其他人已经发表或撰写过的研究成果，也不包含为获得商洛学院或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：_____ 日期：_____

关于论文使用授权的说明

本人完全了解商洛学院有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

☐公开 ☐保密（_____年_____月）（保密的学位论文在解密后应遵守此协议）

签名：

导师签名：

日期：

基于朴素贝叶斯算法的垃圾邮件过滤

摘 要：电子邮件以简单、快捷、低成本、便利等优点为人们带来方便，但同时也造成了垃圾邮件的泛滥。垃圾邮件可能会带来病毒使计算机中毒瘫痪，同时恶化互联网环境，滥用网络带宽与计算机存储资源。因此，提高垃圾邮件过滤技术，已经变得极为紧迫。

为了遏制垃圾邮件的传播，基于朴素贝叶斯算法来帮助用户过滤垃圾邮件。首先将收集一定数量的邮件作为样本邮件进行训练，并充分利用 Python 的文本处理能力将收集到的邮件内容切分成词向量，然后利用词向量出现的概率计算该封邮件是垃圾邮件的概率，对邮件进行分类。当邮件分类为垃圾邮件时，为避免分类错误，将正常邮件分类为垃圾邮件给用户造成损失，本过滤系统设计了词云来显示邮件的关键词，使用户能够迅速了解邮件的主要内容，来减少损失。利用朴素贝叶斯原理对邮件样本进行分类并实现垃圾邮件的过滤，反映该方法对垃圾邮件拦截的有效性，帮助用户远离垃圾邮件。

关键词：垃圾邮件；朴素贝叶斯算法；贝叶斯过滤

Spam Filtering Based on Naive Bayesian Algorithm

Abstract: E-mail brings convenience to people with the advantages of simple, fast, low cost and convenience, but it also causes the flood of spam. Spam can cause viruses to paralyze computers, worsen the Internet environment, and abuse network bandwidth and computer storage resources. Therefore, improving spam filtering technology has become extremely urgent.

In this paper, in order to curb the spread of spam, naive bayesian algorithm is used to help users filter spam. Firstly, a certain number of emails were collected as sample emails for training, and the text processing ability of Python was fully utilized to divide the collected email content into word vectors. Then, the probability of the occurrence of word vectors was used to calculate the probability that the email was spam, and the mail was classified. When the mail is classified as spam, in order to avoid the classification error, the normal mail is classified as spam to the loss of users, the filtering system designed the word cloud to display the key words of the mail, users can quickly understand the main content of the mail, to reduce the loss. The naive bayes principle is used to classify mail samples and realize spam filtering, which reflects the effectiveness of this method to spam interception and helps users stay away from spam.

Key words: Spam; Naive bayesian algorithm; Bayesian filtering

目 录

1 绪论.....	1
1.1 垃圾邮件的危害.....	1
1.2 过滤垃圾邮件的技术现状.....	1
1.3 研究的目的和意义.....	3
2 系统简介.....	3
2.1 贝叶斯原理.....	3
2.2 贝叶斯技术.....	3
2.3 朴素贝叶斯算法.....	4
2.4 训练算法.....	7
2.4.1 计算先验概率.....	7
2.4.2 计算每个词语在各类别邮件中的概率.....	7
2.5 算法修正.....	8
3 系统设计与实现.....	8
3.1 总体设计流程.....	8
3.2 样本数据准备.....	9
3.3 分词.....	9
3.4 生成词汇表.....	10
3.5 生成词向量.....	10
3.6 词云.....	11
4 测试环境.....	11
4.1 Python 3.7 环境.....	11
4.2 Pycharm 开发工具.....	12
5 实验结果及分析.....	13
参考文献.....	15
致谢.....	16
附录.....	17

1 绪论

1.1 垃圾邮件的危害

信息技术的发展使电子邮件成为互联网信息传播和通信的主要方式，具有快速，方便，低成本的优点。虽然电子邮件有利于生活，但垃圾邮件的泛滥也给电子邮件的用户带来了许许多多问题和麻烦。大量的非正常邮件给人们的生活带来的很大的问题，这可能导致邮箱的用户变少，因此让很多研究人员的对这个问题进行了深入的研究。根据中国的反垃圾邮件的第一次调查报告显示，在 2007 年下旬至 2008 年上旬内，垃圾邮件的比例由 60.97%上升到了 63.79%在中国邮箱用户收到邮件中的，上升了 2.82 %，大大高于之前所调查的结果。非正常邮件的问题，对邮箱用户的工作，生活，有不可挽回的损失^[1]。

严重的垃圾邮件问题主要有以下两个原因：

垃圾邮件中广告居多，发送者只用获得邮箱用户的列表，就可以把垃圾邮件发送到用户的邮箱中。

目前没有很有效的限制垃圾邮件的技术使，这使得电子邮箱的用户经常受到垃圾邮件的烦扰。所以，有效的垃圾邮件过滤技术来解决垃圾邮件给用户带来的困扰是非常必要的。

1.2 过滤垃圾邮件的技术现状

垃圾邮件问题越来越严重，人们开始从不同的角度寻找解决方案，目前已经在世界多个地方建立了各种组织来缓解垃圾邮件所带来的困扰。在与垃圾邮件的持续斗争中，一些技术不断改进，新的垃圾邮件的特征不断被开发出来。在垃圾邮件的过滤方面，对此目前主要有以下几种解决办法：使用 IP 地址对垃圾邮件进行过滤，读取邮件的内容对垃圾邮件进行过滤及判断发送邮件者的行为对垃圾邮件进行过滤。

使用 IP 地址对垃圾邮件进行过滤的方法：①黑白名单 ②安全认证法③监测邮箱服务器端^[2]。

读取邮件的内容对垃圾邮件进行过滤包含有使用规则的内容进行的过滤和使用概率统计的内容进行的过滤：

①使用规则的内容：

a.对关键字进行匹配：此技术对邮件的主题进行监测，匹配邮件的主题与垃圾邮件的数据库的常用词汇。如果得到高于阈值的结果，则认定此邮件为垃圾邮件，并将这封

邮件过滤掉，以便不能将其发送给电子邮件的用户。如果得到低于阈值的结果，则确定为普通邮件并且可以正常发送。利用关键字进行判断的优点是该方法相对容易判断，判断的速度也较快，但缺点是误报率较高，给电子邮件用户带来损失^[3]。

b.采用 RoughSet: RoughSet 是研究不确定、不完整的知识的重要的方法。RoughSet 的核心是对多个值的属性的向量的集合进行考虑。它在集合的顶部构建等价的关系，然后确定分类的最小的对象集和最大的对象集。在粗糙集理论中，决策表也叫做为信息系统。属性分为要求属性和决定属性。利用粗糙集对垃圾邮件进行过滤，据实验测试可以达到 80% 的正确率^[4]。

c.采用决策树：数据进行分类的一种重要方法就是决策树对。该过程采用树的形式。形成一个“问题 - 判断 - 问题”的树。决策树的节点具有分支节点和叶节点。将树转到另一个分支，最后转到没有分支的叶节点，叶节点指示相应的类别。过滤垃圾邮件时采用决策树可以达到非常好的过滤结果，但是不能直接采用决策树方法去过滤垃圾邮件，他需要使用其他一些辅助方法^[5]。

②使用概率统计的内容进行的过滤的方法：

根据概率统计过滤内容是指将文本计算为垃圾邮件以过滤垃圾邮件的概率。朴素贝叶斯文本分类算法用作过滤垃圾邮件是目前最好也是最常用的。贝叶斯分类器的原理是以令牌为单位算出文本内容是属于垃圾邮件还是属于普通邮件的概率，对邮件进行判别并分类^[6]。

判断发送邮件者的行为对垃圾邮件进行过滤的方法：

①对发送邮件的工具进行识别:基本原则是根据邮件头中显示的发送工具或发件人的字段信息确定批量发送工具是否发送邮件。如果发现发送的邮件是用批量发送的工具发送的，就将其判断为垃圾邮件并被阻止，使其不能发送到邮箱用户的邮箱中，如果发现无异常，则将其放行。此方法的优势是它可以直接匹配出邮件头中的某些词语是否属于常见的垃圾邮件中的词语但它的缺点也是非常大的，它需要不断地去更新批量发送软件工具和工具的功能；同时，一些普通的电子邮件如果也通过群发发送，如公司给员工群发送信息或公告等,也会拦截，如此就会给邮箱用户带来不便。

②利用流量去控制垃圾邮件的发送:基本方法是根据电子邮箱的管理者去监控电子邮件。根据邮件的内容，电子邮箱的管理者给发送垃圾邮件者进行警告或拒绝它使用电子邮箱。该技术需要对现有邮件网络进行大量更改，效果显而易见,但也是存在误判和难以忍受的危害^[7]。

1.3 研究的目的和意义

目前，垃圾邮件泛滥，给人们的生活带来了诸多麻烦，基于目前过滤垃圾邮件的方法，本文采用了朴素贝叶斯垃圾邮件过滤算法，它的正确率是最高的。朴素贝叶斯算法过滤垃圾邮件可以在样本数据较少的情况下仍然有效，也可以有效的处理多类别问题。

2 系统简介

2.1 贝叶斯原理

垃圾邮件的过滤算法中贝叶斯理论被大量的应用，具体是把它看作是一个分类问题，第一是我们要收集大量的邮件作为邮件样本，第二我们要对收集到的邮件样本进行训练，第三利用训练好的贝叶斯分类器对其他的邮件进行分类。通过训练样本邮件，贝叶斯分类器可以获取所有垃圾邮件的所有特征，并根据垃圾邮件的特征来计算邮件文本属于垃圾邮件或正常邮件的概率，将想要分类的邮件利用邮件特征计算它的概率，基于它的概率把它分到最大的类别中去,来比较精确地概率对实现垃圾邮件的过滤^[8]。

2.2 贝叶斯技术

随机试一个验 E 的可能出现的情况组成的结果是 E 的样本空间将其标记为 S 。样本空间中 E 的每个结果，即在样本空间中的所有元素，称为一个样本点^[9]。随机试验 E 的样本空间 S 的子集都是 E 的一个随机事件。表示在事件 E 发生情况下，某一个的结果 A 可能的发生的概率的值^[10]。

假定 A, B 都是随机的事件，且事件 A 发生的概率大于零，即有：

$$P(B|A) = \frac{P(AB)}{P(A)} \quad \text{公式 (2-1)}$$

在上式中在事件 A 发生的条件下事件 B 发生的条件概率为 $P(B|A)$ ^[11]。

假定事件 A 发生的概率大于零，则有

$$P(AB) = P(B|A)P(A) \quad \text{公式 (2-2)}$$

上式是著名的乘法公式，意为两个随机事件的乘积可能出现的概率。

假定随机事件 E 的样本空间是 S ，并且 A 是事件 E 的子事件， B_1, B_2, \dots, B_n 为样本空间 S 的一个划分，且 $P(B_i) > 0 (i=1, 2, \dots, n)$ ，则有

$$P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \dots + P(A|B_n)P(B_n) \quad \text{公式 (2-3)}$$

将公式 (2-3) 称为全概率公式^[12]。

定理 2.3：试验 E 的样本空间为 S ， A 为 试验 E 的事件,且 $P(A_i) > 0$ ，

$P(B_i) > 0 (i=1,2,\dots,n)$, 则有

$$p(B_i | A) = \frac{P(A | B_i)P(B_i)}{\sum_{j=1}^n p(A | B_j)P(B_j)} \quad \text{公式 (2-4)}$$

上述公式是著名的贝叶斯（Bayes）公式^[13]。在贝叶斯理论中，依据已有资料来确定的事件发生的概率我们称之为先验概率^[14]，其中，概率值的大小取决于已有资料，它是指计算前的概率。

基于贝叶斯公式及先验概率对将要判定的事件做出的估计我们称之为后验概率。要进行后验概率估计的前提是必须要计算先验概率并且其是有效的且充分的。

通过已有资料和计算方法得到的先验概率是贝叶斯概率，并运用事件发生的可能性的方法对后验概率进行的猜测。

2.3 朴素贝叶斯算法

目前，在关于垃圾邮件内容过滤的算法中，我们使用最多，最为广泛的算法是基于朴素贝叶斯的文本分类算法^[15]。其中，朴素贝叶斯文本分类算法是通过计算目标属于某个类别的概率，最后概率最大的一类即是文本的类别。在“贝叶斯假设”的基础上引入了朴素贝叶斯的分类的算法。朴素贝叶斯的分类的算法是假设实验的目标所有的特征都互不影响，朴素贝叶斯的分类的算法的基础上用来简化概率的计算。

贝叶斯的文本的分类的算法的基本原理如公式 2-5 所示：

$$P(c_j | d_x) = \frac{p(c_j)p(d_x | c_j)}{p(d_x)}, j = 1, 2, \dots, |c| \quad \text{公式 (2-5)}$$

公式 2-6 为全概率公式：

$$p(d_x) = \sum_{j=1}^{|c|} p(c_j)p(d_x | c_j) \quad \text{公式 (2-6)}$$

由训练集进行计算 c_j 类的先验概率如公式 2-7 所示：

$$p(c) = \frac{\text{训练集中属于 } c_j \text{ 类的文本数量}}{\text{训练集中的文本总数量}} \quad \text{公式 (2-7)}$$

由文本中出现的特征的条件概率可计算公式 2-7 条件概率 $P(d_x | c_j)$ 。

基于朴素贝叶斯的文本的分类的算法有两种基本的模型：A.多项式事件模型^[16]。B.多变量贝努里事件模型^[17]。

(1) 多项式事件模型和多变量贝努里事件模型的区别主要在于：在公式（2-5）中

以不一样的计算方法来计算 $P(d_x | c_j)$ 。

在多变量贝努里事件模型中，样本内容中的分词的每一个分词都有权重。假设有 n 个分词，把样本内容的分词看做是一个事件，文本是通过分词结果产生的，即目标词是否出现。如特征 w_i 在文本 dx 中的出现情况为 $B_{xt} = 1 / 0$ 为， B_{xt} 表示是否出现则有公式 2-8:

$$p(d_x | c_j) = \prod_{t=1}^n (B_{xt} P(w_t | c_j)) + (1 - B_{xt})(1 - P(w_t | c_j)) \quad \text{公式 (2-8)}$$

在公式 2-8 中文本属于 C_j 类的情况下 W_t 特征的可能性为 $P(W_t | C_j)$ 。由公式 2-8 可以得到，在多变量的贝努里事件的模型中，所有单个分词的概率之积就是样本邮件的判断依据。如果文本之中出现目标分词，则最后结果为： $P(W_t | C_j)$ ，如果不出现，乘的项是 $1 - P(W_t | C_j)$ 。 $P(W_t | C_j)$ 的计算方式为公式 2-9 所示：

$$P(W_t | C_j) = \frac{C_j \text{类中特征 } W_t \text{ 在其中出现的文本数量}}{C_j \text{类的文本数量}} \quad \text{公式 (2-9)}$$

其中平滑的处理过程如公式 2-10 所示：

$$P(W_t | C_j) = \frac{1 + C_j \text{类中特征 } W_t \text{ 在其中出现的文本数量}}{2 + C_j \text{类的文本数量}} \quad \text{公式 (2-10)}$$

由以上可得多变量贝努里事件模型有以下两个的特征：

① 不需分词在样本中的封数，在计算 $P(d_x | c_j)$ 和 $P(W_t | C_j)$ 时；

② 没有出现在目标文本中的词，计算 $P(d_x | c_j)$ 时乘以项 $1 - P(W_t | C_j)$ 。在进行判别邮件内容时，只用虑垃圾 SPAM 和非垃圾邮件 HAM 这两种情况，设 $C=0$ 表示正常邮件， $C=1$ 表示非正常邮件，则可得到公式 2-11 结果：

$$p(c = 1 | d_x) = \frac{p(c = 1) p(d_x | c = 1)}{p(d_x)} \quad \text{公式 (2-11)}$$

由以上的公式 (2-9) 和公式 (2-8) 可可有如下结果：

$$P(d_x | c = 1) = \prod_{t=1}^n B_{xt} P(w_t | c = 1) + (1 - B_{xt})(1 - P(w_t | c = 1)) \quad \text{公式 (2-12)}$$

$$P(d_x | c = 0) = \prod_{t=1}^n B_{xt} P(w_t | c = 0) + (1 - B_{xt})(1 - P(w_t | c = 0)) \quad \text{公式 (2-13)}$$

$$P(d_x) = P(c = 1) P(d_x | c = 1) + (1 - P(c = 1)) P(d_x | c = 0) \quad \text{公式 (2-14)}$$

对样本文件进行计算 $P(c=1)$ 、 $P(w_t|c=1)$ 和 $P(w_t|c=0)$ ，判别时由以上公式计算即

可知道目标文本分类正确与否。

(2) 在多项式事件模型中, 对评判结果造成影响的一个重要原因是样本内容的分词在邮件中出现的封数。如果样本中的分词是彼此不重复的。设 d_x 为样本中包含的分词数, 由分词在样本中的出现封数的和, 文本中分词的数量为 n , 可得 $P(d_x | c_j)$ 适合该模型, 可得公式 2-15 结果:

$$P(d_x | c_j) = P(|d_x|) \times |d_x|! \times \prod_{t=1}^n \frac{P(w_t | c_j)^{N_{xt}}}{N_{xt}!} \quad \text{公式 (2-15)}$$

计算训练集时如公式 2-16 所示:

$$P(w_t | c_j) = \frac{c_j \text{ 类的所有文本中特征词 } w_t \text{ 的出现次数}}{c_j \text{ 类的所有文本中出现的特征词总数}} \quad \text{公式 (2-16)}$$

平滑处理如公式 2-17 所示:

$$P(w_t | c_j) = \frac{1 + c_j \text{ 类的所有文本中特征词 } w_t \text{ 的出现次数}}{n + c_j \text{ 类的所有文本中出现的特征词总数}} \quad \text{公式 (2-17)}$$

在对邮件进行分类时, 只需判断邮件是正常的还是非正常的, 假定 $C=0$ 表示正常邮件, $C=1$ 表示非正常邮件, 由公式(2-15)和公式(2-16)可以得到公式 2-18 的结果:

$$\begin{aligned} P(c=1 | d_x) &= \frac{P(c=1)P(d_x | c=1)}{P(c=1)P(d_x | c=1) + P(c=0)P(d_x | c=0)} \\ &= \frac{1}{1 + \frac{1 - P(c=1)}{P(c=1)} \times \frac{P(d_x | c=0)}{P(d_x | c=1)}} \end{aligned} \quad \text{公式 (2-18)}$$

对同一封邮件, 由公式 (2-18) 得:

$$\frac{P(d_x | c=0)}{P(d_x | c=1)} = \prod_{t=1}^n \frac{P(w_t | c=0)^{N_{xt}}}{P(w_t | c=1)^{N_{xt}}} \quad \text{公式 (2-19)}$$

朴素贝叶斯文本分类器是对文本的内容进行判断并计算它出现的可能性, 并计算它出现的可能性, 将所得概率与设定的阈值相比较, 判别目标内容是有用的还是无用的。阈值是概率计算中非常重要的一个参数。垃圾邮件与样本的相像程度阈值就大一些。将一些与垃圾邮件相似程度低的邮件判定为正常邮件, 这种情况下能减少判断的情况, 但是, 垃圾邮件与样本邮件的相似度低, 阈值就会小一些, 表示将与垃圾邮件相像程度小的垃圾邮件正常发送, 但是这种情况下可能把正常邮件误判为垃圾邮件的概率会变大, 一旦判断错误那么就会对邮箱用户带来很多的损失。因此可以通过不断地训练文本来调整阈值的大小^[18]。

2.4 训练算法

2.4.1 计算先验概率

先验概率是指根据以往经验和分析得到的概率，如全概率公式，它往往作为“由因求果“问题中的”因“出现的概率。先验概率的分类利用过去历史资料计算得到的先验概率，称为客观先验概率；当历史资料无从取得或资料不完全时，凭人们的主观经验来判断而得到的先验概率，称为主观先验概率^[19]。

先验概率的条件：先验概率是通过古典概率模型加以定义的，故又称为古典概率。古典概率模型要求满足两个条件^[20]：(1)试验的所有可能结果是有限的；(2)每一种可能结果出现的可能性(概率)相等。若所有可能结果的总数为 N ，随机事件 A 包括 n 个可能结果，那么随机事件 A 出现的概率为 n/N 。

在贝叶斯统计中，先验概率分布，即关于某个变量 p 的概率分布，是在获得某些信息或者依据前，对 p 的不确定性进行猜测。例如， p 可以是抢火车票开始时，抢到某一车次的概率。这是对不确定性（而不是随机性）赋予一个量化的数值的表征，这个量化数值可以是一个参数，或者是一个潜在的变量。

先验概率仅仅依赖于主观上的经验估计，也就是事先根据已有的知识的推断，在应用贝叶斯理论时，通常将先验概率乘以似然函数再归一化后，得到后验概率分布，后验概率分布即在已知给定的数据后，对不确定性的条件分布。

贝叶斯估计，是在给定训练数据 D 时，确定假设空间 H 中的最佳假设^[21]。最佳假设：一种方法是把它定义为在给定数据 D 以及 H 中不同假设的先验概率的有关知识下的最可能假设。贝叶斯理论提供了一种计算假设概率的方法，基于假设的先验概率、给定假设下观察到不同数据的概率以及观察到的数据本身。

使用贝叶斯算法的本质是由概率来判断一件事出现的概率的一种方式，使用出现可能性大的当作分类的依据，这是贝叶斯算法的中心的内容，其中最能代表它的是对事件可能发生的可信度来进行分析。置信度是事件的概率的一种：先验概率是人们由自己了解的知识对事件可能发生概率一种评估。

2.4.2 计算每个词语在各类别邮件中的概率

要分类目标邮件是否为垃圾邮件，首先必须得到本邮件中每个词在垃圾邮件中的概率。

在垃圾邮件的条件下第 i 个特征的概率，首先先将所有的类别为 1 的词向量相加，

可以得到每个特征的个数，所以再除以在类别为 1 的单词总数得到的就是在垃圾邮件中每个单词的概率了。在得到每个特征的概率后，将所有特征相乘，可得到目标邮件是垃圾邮件和非垃圾邮件的概率。

2.5 算法修正

在进行分类时，多个概率乘积得到类别，但是如果有一个概率为 0，则最后的结果为 0，因此为了避免未出现的属性值，在估计概率时同必须要进行“平滑”处理，常用的是平滑处理是使用拉普拉斯修正^[22]（Laplacian correction）具体来说，在计算的时候，将分子加 1，分母加上类别数 N 同样在计算的时候在分子加 1，分母加 2，这样就避免了整个算式为 0 的特殊情况^[20]。当值过小时可能会出现相乘溢出，在实际中对概率取对数的形式，再进行相乘，可以防止相乘结果溢出的现象。

3 系统设计与实现

3.1 总体设计流程

设计思路如流程图所示，首先是对收集到的样本邮件内容进行分词，将每封邮件的分词进行并集组成集合，并将所有集合组成词汇表，生成词向量，利用这些词向量计算每个词出现的概率以进行算法的训练之后再判断一封邮件是否是垃圾邮件若是垃圾邮件用词云展示它的主要内容以避免分类错误。流程图如 3-1 图所示：

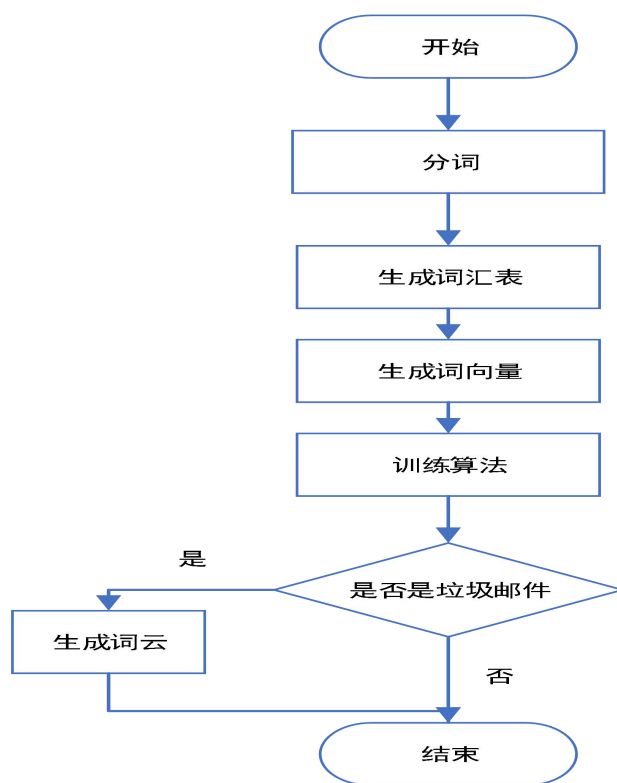


图 3-1 垃圾邮件过滤算法基本思想

利用流程图来展现设计思路能够使阅读者迅速的了解本设计系统的主要过程。

3.2 样本数据准备

垃圾邮件一般具有批量发送的特征。其内容包括赚钱信息、成人广告、商业或个人网站广告、电子杂志、连环信等。垃圾邮件可以分为良性和恶性的。良性垃圾邮件是各种宣传广告等对收件人影响不大的信息邮件。恶性垃圾邮件是指具有破坏性的电子邮件。例如具有攻击性的广告：夸张不实，包括情色、钓鱼网站。

一些有心人会从网上多个 BBS 论坛、新闻组等收集网民的电子邮件地址，再售予广告商，从而发送垃圾邮件到这些地址。在这些邮件，往往可找到从收信人的清单移除的连结。当使用者依照连结指示去做时，广告商便知道这地址有效，使用者便会收到更多垃圾邮件。

随着垃圾邮件的问题日趋严重，多家软件商也各自推出反垃圾邮件的软体。但垃圾邮件的格式更加日新月异，以避过此类软体的侦测。垃圾邮件的攻击有些垃圾邮件发送组织或是非法信息传播者，为了大面积散布信息，常采用多台机器同时巨量发送的方式攻击邮件服务器，造成邮件服务器大量带宽损失，并严重干扰邮件服务器进行正常的邮件递送工作。

基于上述垃圾邮件的特征，我下载了公开数据，公开数据包含垃圾邮件 250 封和正常邮件 250 封，在代码实现中将其作为邮件的训练的样本。

3.3 分词

要使用朴素贝叶斯基于邮件内容实现垃圾邮件处理就必须计算每个词的概率所以首先要对邮件内容进行分词才能计算每个分词的概率。分词是把邮件内容划分成一个个单词的形式，此时可以想到用正则表达式来分词，但正则表达式不能很好的分割中文，所以这里选用的是 python 的第三方库 jieba 来实现邮件内容分割成单个词语。

jieba 是优秀的中文分词，第三方库中文文本需要通过分词获得单个的词语，jieba 是优秀的中文分词第三方库，需要额外安装 jieba 库提供三种分词模式，最简单只需掌握一个函数。

jieba 分词的原理：Jieba 分词依靠中文词库，利用一个中文词库，确定汉字之间的关联概率，汉字间概率大的组成词组，形成分词结果，除了分词，用户还可以添加自定义的词组。

jieba 分词有三种分词模式：精确模式、全模式、搜索引擎模式。

精确模式：把文本精确的切分开，不存在冗余单词；

全模式：把文本中所有可能的词语都扫描出来，有冗余；

搜索引擎模式：在精确模式基础上，对长词再次切分。

代码示例如图 3-2 所示：

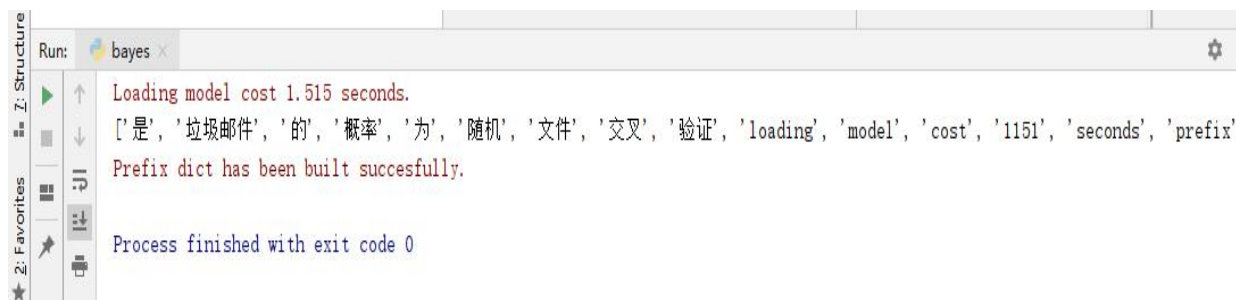
```

116 def textParse(bigString):
117     """
118     函数说明:字符串解析函数,可处理中文和英文(分词)
119     """
120     import re
121     import jieba
122     # 切分文本
123     listOfTokens = jieba.lcut(bigString)
124     # 去掉标点符号
125     newList = [re.sub(r'\W*', '', s) for s in listOfTokens]
126     # 删除长度为0的空值
127     return [tok.lower() for tok in newList if len(tok) > 0]

```

图 3-2 分词示例

分词结果如图 3-3 所示：



```

Run: bayes x
Loading model cost 1.515 seconds.
['是', '垃圾邮件', '的', '概率', '为', '随机', '文件', '交叉', '验证', 'loading', 'model', 'cost', '1151', 'seconds', 'prefix']
Prefix dict has been built successfully.
Process finished with exit code 0

```

图 3-3 分词结果

3.4 生成词汇表

为了更加直观的统计每个分词结果在训练的样本文件内出现的个数，必须将所有的样本词汇组合生成一个词汇表。

将所有的邮件内容进行分词后生成一个集合，在这个集合内，每个样本单词只出现一次，词汇表是一个列表形式如：["cute", "love", "help", "garbage", "quit"...]。生成词汇表是为后续生成词向量做准备。

3.5 生成词向量

在代码实现过程中，每一封邮件的词汇都存在于词汇表中，因此可以将每一封邮件

的邮件内容分词后生成一个词向量，存在几个则为几，不存在为 0，例如：["love", "garbage"], 则他的词向量为 [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。

生成词向量的作用是在对一封邮件分类时就可直接相加他们出现次数的和即 1 的数目，然后除以总词数的和，以此得到每个词的概率。

3.6 词云

词云最早是来自里奇·戈登，他曾是美国西北大学的新闻学副教授、新媒体专业主任。词云就是对文本中出现次数较多的词语给予以在视觉上的突出的显示，形成一种类似于"关键词的渲染"，从而让阅读者快速滤掉大量的无用的信息的效果，让浏览的人只需粗略扫过词云就可以领略文本的主要意思。在文本分类结束后，为了防止把正常的邮件错误分类为了垃圾邮件，在分类结果为垃圾邮件时，通过突出展现邮件关键词，以避免此问题。



图 3-4 词云图示例

4 测试环境

4.1 Python 3.7 环境

Python 是解释型的脚本语言，可以应用在很多领域：Web 开发、科学计算、统计、教育、桌面界面开发、软件开发、后端开发等。使用 Python 做科学计算的优点有很多：

- (1) 完全免费。
- (2) 许多科学计算都提供 Python 接口，可以很方便的调用。
- (3) 相较其他软件来说 Python 是一种非常容易学习和严谨的计算机的语言，其代码非常易读并易于维护。
- (4) 非常易于操作文本文件；使用非常广泛，存在大量的开发文档。

(5) 有着非常丰富的扩展库，可以非常容易用来完成高级的任务，程序的各种功能会很容易实现。

4.2 Pycharm 开发工具

PyCharm 是一种 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。PyCharm 是一个非常好用的编译工具对于 Python 来说，有很多帮助对于使用者来说，使用者在使用 Python 语言进行程序编写时，可大大的增加开发效率。另外，PyCharm 还提供了一些很好的功能用于 Django 开发，同时支持 Google App Engine，更酷的是，PyCharm 支持 IronPython。其提供了一个带编码补全，代码片段，支持代码折叠和分割窗口的智能、可配置的编辑器，可帮助用户更快更轻松地完成编码工作。该 IDE 可帮助用户即时从一个文件导航至另一个，从一个方法至其申明或者用法甚至可以穿过类的层次。若用户学会使用其提供的快捷键的话甚至能更快。

用户可使用其编码语法，错误高亮，智能检测以及一键式代码快速补全建议，使得编码更优化。有了该功能，用户便能在项目范围内轻松进行重命名，提取方法/超类，导入域/变量/常量，移动和前推/后退重构。有了它自带的 HTML，CSS 和 JavaScript 编辑器，用户可以更快速的通过 Django 框架进行 Web 开发。这里可选择使用 Python 2.5 或者 2.7 运行环境，为 Google App 引擎进行应用程序的开发，并执行例行程序部署工作。比如在程序中间进行调试时，运行界面如图 5-1 所示。

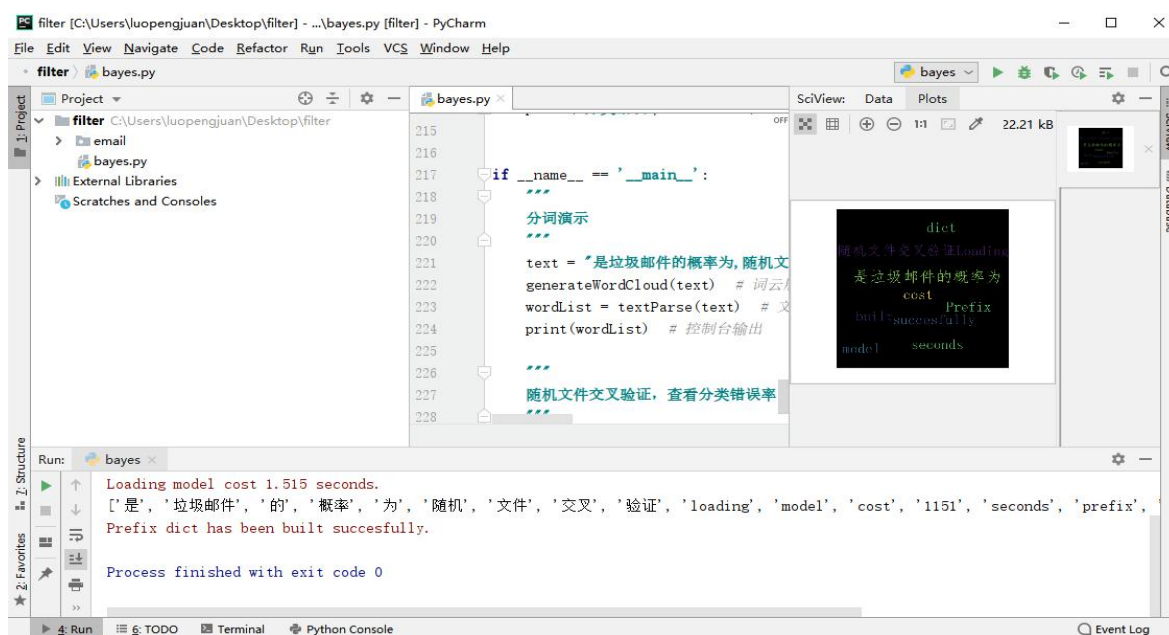
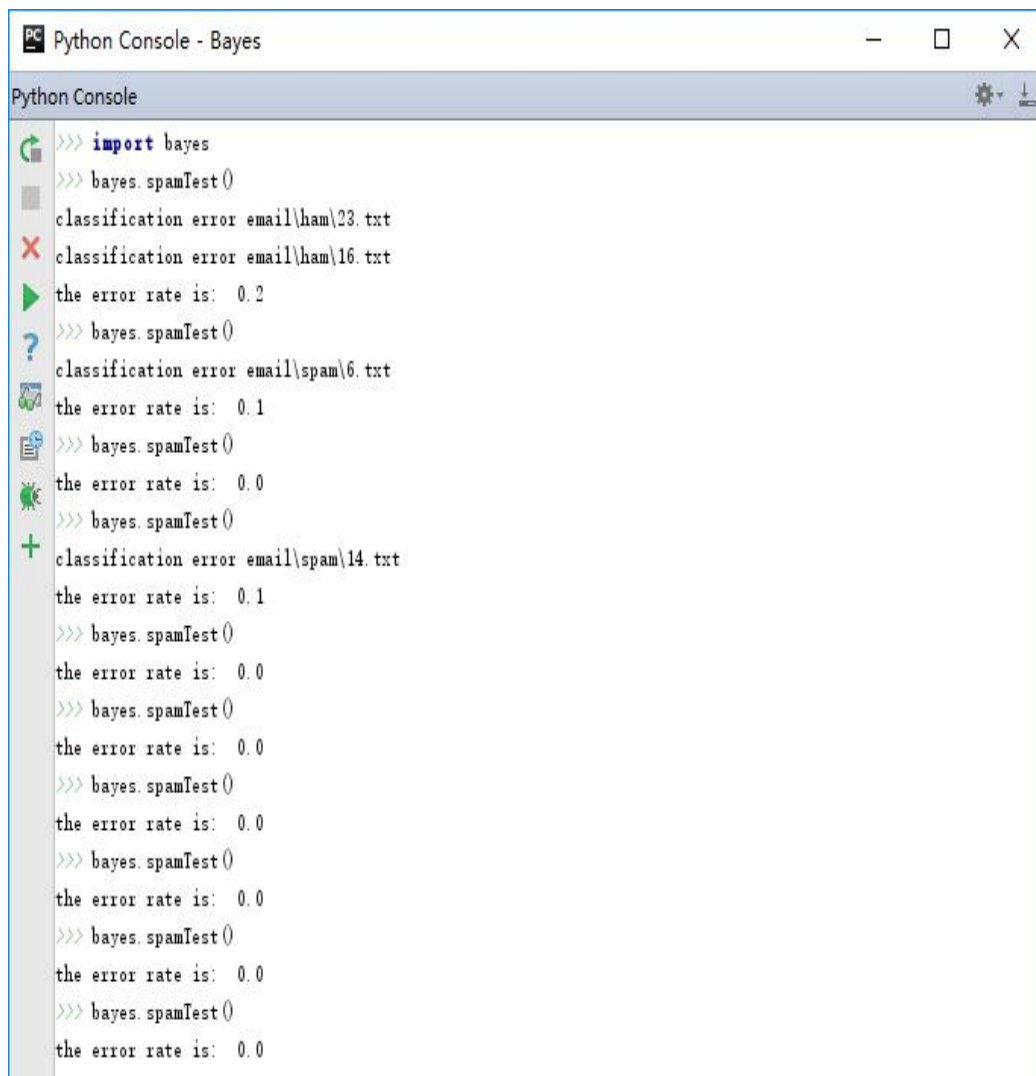


图 5-1pycharm 运行界面图

5 实验结果及分析

这里首先将 500 封邮件读进列表中，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，随机的从样本中选择 150 个样本进行测试，350 个进行训练。

第三阶段应用阶段调用第三方库对邮件内容进行分词，去重，再计算单个词语为垃圾邮件词语的概率及各个词语的联合概率，进行判别是否为垃圾邮件。



```
>>> import bayes
>>> bayes.spamTest()
classification error email\ham\23.txt
classification error email\ham\16.txt
the error rate is: 0.2
>>> bayes.spamTest()
classification error email\spam\6.txt
the error rate is: 0.1
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
classification error email\spam\14.txt
the error rate is: 0.1
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
>>> bayes.spamTest()
the error rate is: 0.0
```

图 5-2 运行结果图

经过 10 次重复交叉验证，最后的正确分类邮件个数 1445，错误分类邮件个数 5 个，其中把正确邮件分类为错误邮件个数为 2，错误邮件分为正确邮件个数为 3 个，由此可知由朴素贝叶斯算法过滤垃圾邮件的正确率在到 99.3%以上。由此分析此系统对垃圾邮件过滤有效。

6 结语

对垃圾邮件进行过滤是全世界一直研究的，因为垃圾邮件的大量传播会给整个互联网造成很大的压力。大量的垃圾邮件会造成的邮箱的服务器的拥堵，并且会极大的占用了邮箱使用者的内存，会占用邮箱用户的时间和精力也可能会对邮箱用户带来金钱上的损失；甚至还有的邮件还会盗用别人的邮箱的地址做为发信的地址，这将非常大地对其他人的信誉造成损坏，这些结果一旦发生最终使得邮箱用户流失。

现代很多人在研究过滤垃圾邮件的技术，也很很多有用的方法，但是垃圾邮件也是依据更新的过滤方法不断地修改垃圾邮件的最新特征,这也使得传统的垃圾邮件过滤的技术无法被及时的发现和检测出来。所以，如果要把垃圾邮件完全阻挡在系统外部，光靠过滤垃圾邮件的手段不能完全解决的，还必须需要有关部门的重视和积极参与，可通过宣传或者立法等办法，通过使用法律的手段来对实现垃圾邮件的制造者进行相关的制裁。所以只有我们每个-个人都自觉行动，加入到抵制垃圾邮件中，并且使用先进的技术武装网络，用法律的规管理制度为从根本上消除垃圾邮件。

本系统使用了基于朴素贝叶斯的方法来过滤邮件，虽然能够在比较大的程度上将垃圾邮件过滤掉，但不能很好的应对变化的邮件特征，希望有关部门能够积极的行动起来，积极加入垃圾邮件过滤，共同维护网络安全中来，从根本上杜绝垃圾邮件。

参考文献

- [1] 周俊怡. 一种混合垃圾邮件过滤技术研究[D].电子科技大学,2009.
- [2] 徐梦龙,黄家旺.朴素贝叶斯算法在垃圾邮件过滤方面的应用[J].网络安全技术与应用,2018(07):46-47.
- [3] 曹翠玲,王媛媛,袁野,赵国冬.用于垃圾邮件的贝叶斯过滤算法研究[J].网络与信息安全学报,2017,3(03):64-70.
- [4] 赵敬慧,魏振钢.改进的贝叶斯垃圾邮件过滤算法[J].计算机系统应用,2016,25(10):137-140.
- [5] 魏如玉. 中文垃圾邮件过滤方法的研究[D].辽宁大学,2016.
- [6] 张凡. 文本分类在垃圾邮件拦截系统中的应用[D].西安电子科技大学,2014.
- [7] 王龙龙. 基于贝叶斯算法的垃圾邮件过滤系统设计与实现[D].吉林大学,2014.
- [8] 欧飒. 一种基于贝叶斯分类的邮件网络协同过滤算法[D].哈尔滨工程大学,2014.
- [9] 杨赫,孙广路,何勇军.基于朴素贝叶斯模型的邮件过滤技术[J].哈尔滨理工大学学报,2014,19(01):49-53.
- [10] 朱强. 贝叶斯算法在智能终端信息过滤中的应用研究[D].中南大学,2013.
- [11] 方鹏. 基于内容分析的垃圾邮件过滤技术的设计与实现[D].电子科技大学,2013.
- [12] 次曲(Tse Qu). 基于朴素贝叶斯算法的藏文垃圾邮件过滤关键技术研究[D].电子科技大学,2013.
- [13] 梁婷. 基于内容的垃圾邮件过滤技术研究[D].华东师范大学,2013.
- [14] 李爽.改进型贝叶斯算法网络垃圾邮件信息过滤技术[J].科技通报,2012,28(04):180-181.
- [15] 钱诚. 改进的贝叶斯分类法在垃圾邮件过滤中的应用研究[D].华东理工大学,2012.
- [16] 马小龙. 一种改进的贝叶斯算法在垃圾邮件过滤中的研究[J]. 计算机应用研究,2012,29(03):1091-1094.
- [17] 陈强. 基于贝叶斯方法的垃圾邮件过滤技术的研究[D].沈阳工业大学,2011.
- [18] Yi Man. Design and Implementation of the OLAP Cache Mechanism Based on Incremental Learning Naive Bayesian Algorithm[A]. IEEE、华中师范大学.Proceedings of 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI 2016) [C].IEEE、华中师范大学:IEEE BEIJING SECTION(跨国电气电子工程师学会北京分会),2016:4.
- [19] Yishan Gong, Qiang Chen. Research of spam filtering based on Bayesian algorithm[P]. Computer Application and System Modeling (ICCASM), 2010 International Conference on,2010.
- [20] 严灼. 基于内容解析的垃圾邮件过滤技术研究[D].安徽理工大学,2011.
- [21] 纪繁芳. 基于贝叶斯过滤的文本分类技术的研究与实现[D].电子科技大学,2011.
- [22] 陈强. 基于贝叶斯方法的垃圾邮件过滤技术的研究[D].沈阳工业大学,2011.

致谢

大学四年学习时光已经接近尾声，在此我想对我的母校，我的父母、亲人们，我的老师和同学们表达我由衷的谢意。感谢我的家人对我大学四年学习的默默支持；感谢我的母校商洛学院给了我在大学四年深造的机会，让我能继续学习和提高；感谢商洛学院的老师和同学们四年来的关心和鼓励。

老师们课堂上的激情洋溢，课堂下的谆谆教诲；同学们在学习中的认真热情，生活上的热心主动，所有这些都让我的四年充满了感动。我做毕业设计的每个阶段，从选题到查阅资料，论文提纲的确定，中期论文的修改，后期论文格式调整等各个环节中都给予了我悉心的指导，在此谨向王博老师致以诚挚的谢意和崇高的敬意。

四年的大学时光匆匆走过，在此，我再次感谢包括此次论文指导老师王博在内的所有商洛学院的教师，感谢你们四年孜孜不倦的教诲。

附录

```
# -*- coding: UTF-8 -*-
import numpy

def generateWordCloud(content):
    """
    函数说明:生成词云
    """
    from matplotlib import pyplot
    from wordcloud import WordCloud
    # 词云参数
    wc = WordCloud(collocations=False, font_path='simfang.ttf', width=1400, height=1400,
margin=2).generate(content)

    pyplot.imshow(wc)
    pyplot.axis("off")
    pyplot.show()

def createVocabList(dataSet):
    """
    函数说明:将切分的实验样本词条整理成不重复的词条列表，也就是词汇表
    Parameters:
        dataSet - 整理的样本数据集
    Returns:
        vocabSet - 返回不重复的词条列表，也就是词汇表
    """
    # 创建一个空的不重复列表
    vocabSet = set([])
    for document in dataSet:
        # 取并集
        vocabSet = vocabSet | set(document)
    return list(vocabSet)
```

```

def setOfWords2Vec(vocabList, inputSet):
    """
    函数说明:根据 vocabList 词汇表, 将 inputSet 向量化, 向量的每个元素为 1 或 0
    Parameters:
        vocabList - createVocabList 返回的列表
        inputSet - 切分的词条列表
    Returns:
        returnVec - 文档向量,词集模型
    """
    # 创建一个其中所含元素都为 0 的向量
    returnVec = [0] * len(vocabList)
    for word in inputSet:
        # 遍历每个词条
        if word in vocabList:
            # 如果词条存在于词汇表中, 则置 1
            returnVec[vocabList.index(word)] = 1
        else:
            print("the word: %s is not in my Vocabulary!" % word)
    # 返回文档向量
    return returnVec

def trainNB0(trainMatrix, trainCategory):
    """
    函数说明:朴素贝叶斯分类器训练函数
    Parameters:
        trainMatrix - 训练文档矩阵, 即 setOfWords2Vec 返回的 returnVec 构成的矩阵
        trainCategory - 训练类别标签向量, 即 loadDataSet 返回的 classVec
    Returns:
        p0Vect - 正常邮件类的条件概率数组
        p1Vect - 垃圾邮件类的条件概率数组
        pAbusive - 文档属于垃圾邮件类的概率
    """
    # 计算训练的文档数目
    numTrainDocs = len(trainMatrix)
    # 计算每篇文档的词条数

```



```

numWords = len(trainMatrix[0])
# 文档属于垃圾邮件类的概率
pAbusive = sum(trainCategory) / float(numTrainDocs)
# 创建 numpy.ones 数组,词条出现数初始化为 1,拉普拉斯平滑
p0Num = numpy.ones(numWords)
p1Num = numpy.ones(numWords)
# 分母初始化为 2 ,拉普拉斯平滑
p0Denom = 2.0
p1Denom = 2.0
for i in range(numTrainDocs):
    if trainCategory[i] == 1:
        # 统计属于侮辱类的条件概率所需的数据, 即  $P(w_0|1), P(w_1|1), P(w_2|1) \dots$ 
        p1Num += trainMatrix[i]
        p1Denom += sum(trainMatrix[i])
    else:
        # 统计属于非侮辱类的条件概率所需的数据, 即  $P(w_0|0), P(w_1|0), P(w_2|0) \dots$ 
        p0Num += trainMatrix[i]
        p0Denom += sum(trainMatrix[i])
# 取对数, 防止下溢出
p1Vect = numpy.log(p1Num / p1Denom)
p0Vect = numpy.log(p0Num / p0Denom)
# 返回属于正常邮件类的条件概率数组, 属于侮辱垃圾邮件类的条件概率数组, 文档属于垃圾
# 邮件类的概率
return p0Vect, p1Vect, pAbusive

def classifyNB(vec2Classify, p0Vec, p1Vec, pClass1):
    """
    函数说明:朴素贝叶斯分类器分类函数
    Parameters:
        vec2Classify - 待分类的词条数组
        p0Vec - 正常邮件类的条件概率数组
        p1Vec - 垃圾邮件类的条件概率数组
        pClass1 - 文档属于垃圾邮件的概率
    Returns:
        0 - 属于正常邮件类
        1 - 属于垃圾邮件类
    """

```

```

"""
p1 = sum(vec2Classify * p1Vec) + numpy.log(pClass1)
p0 = sum(vec2Classify * p0Vec) + numpy.log(1.0 - pClass1)
if p1 >= p0:
    return 1, p1 / (p1 + p0)
else:
    return 0, p0 / (p1 + p0)

```

```
def textParse(bigString):
```

```

"""
函数说明:字符串解析函数，可处理中文和英文（分词）
"""
import re
import jieba
# 切分文本
listOfTokens = jieba.lcut(bigString)
# 去掉标点符号
newList = [re.sub(r'\W*', "", s) for s in listOfTokens]
# 删除长度为 0 的空值
return [tok.lower() for tok in newList if len(tok) > 0]

```

```
def readLearnFile():
```

```

"""
函数说明:读取样本文件
"""
docList = []
classList = []
fileNameList = []
# 遍历 250 个 txt 文件
for i in range(1, 251):
    # 读取每个垃圾邮件，并字符串转换成字符串列表
    fileWordList = textParse(open('email/spam/' + str(i), 'r').read())
    docList.append(fileWordList)
    fileNameList.append('spam/' + str(i))
# 标记垃圾邮件，1 表示垃圾文件

```

```

classList.append(1)
# 读取每个非垃圾邮件，并字符串转换成字符串列表
fileWordList = textParse(open('email/ham/' + str(i), 'r').read())
# 读取每个非垃圾邮件，并字符串转换成字符串列表
docList.append(fileWordList)
# 标记正常邮件，0 表示正常文件
classList.append(0)
fileNameList.append('ham/' + str(i))
return docList, classList, fileNameList

```

def selectTestFile():

"""

函数说明:随机选取 150 个文件测试

Returns:

trainingSet - 训练文件索引

testSet - 测试文件索引

"""

import random

训练文件索引

trainingSet = list(range(500))

测试文件索引

testSet = []

从 500 个邮件中，随机挑选出 350 个作为训练集,150 个做测试集

for i in range(150):

随机选取索引值

randIndex = int(random.uniform(0, len(trainingSet)))

添加测试集的索引值

testSet.append(trainingSet[randIndex])

在训练集列表中删除添加到测试集的索引值

del (trainingSet[randIndex])

return trainingSet, testSet

def randFileTest():

"""

函数说明:测试朴素贝叶斯分类器，使用朴素贝叶斯进行交叉验证

```

"""
# docList-读样本邮件后返回的单词列表
# classList-读样本邮件的向量列表（0 1）
# fileNameList-读样本邮件的文件名列表（0 1）
docList, classList, fileNameList = readLearnFile()
# 创建不重复的词汇表
vocabList = createVocabList(docList)
# 从 500 个邮件中，随机挑选出 350 个作为训练集,150 个做测试集
trainingSet, testSet = selectTestFile()
# 创建训练集矩阵和训练集类别标签系向量
trainMat = []
trainClasses = []
# 遍历训练集
for docIndex in trainingSet:
    # 将生成的词集模型添加到训练矩阵中
    trainMat.append(setOfWords2Vec(vocabList, docList[docIndex]))
    # 将类别添加到训练集类别标签系向量中
    trainClasses.append(classList[docIndex])
# 训练朴素贝叶斯求取邮件类别的概率
p0V, p1V, pSpam = trainNB0(numpy.array(trainMat), numpy.array(trainClasses))
# 错误分类计数
errorCount = 0
# 遍历测试集
for docIndex in testSet:
    # 测试集的词集模型
    wordVector = setOfWords2Vec(vocabList, docList[docIndex])
    # 验证分类是否错误
    isSpan, pValue = classifyNB(numpy.array(wordVector), p0V, p1V, pSpam)
    if isSpan != classList[docIndex]:
        # 错误计数加 1
        errorCount += 1
        print("分类错误: ", fileNameList[docIndex])
        generateWordCloud(open('email/' + fileNameList[docIndex]).read())
    else:
        print("分类正确: ", fileNameList[docIndex])
print('分类错误率: %.2f % (float(errorCount) / len(testSet))

```

```
if __name__ == '__main__':  
    """  
    分词演示  
    """  
    # text = "是垃圾邮件的概率为,随机文件交叉验证 Loading model cost 1.151      seconds.Prefix  
dict has been built succesfully."  
    # generateWordCloud(text) # 词云展示  
    # wordList = textParse(text) # 文字分割  
    # print(wordList) # 控制台输出  
  
    """  
    随机文件交叉验证, 查看分类错误率  
    """  
    randFileTest()
```