**《大数据分析与应用》**

**课程作业论文**

|  |  |  |
| --- | --- | --- |
| **论 文 题 目：** | 基于微博的情感分析方法对比 | |
| **成 员 及 学 号：** | 余维航 | 3220180896 |
| 郑泉斌 | 3220180906 |
| 柯楚雯 | 3220180816 |
| **学 院：** | 计算机学院 | |
| **指 导 老 师：** | 张华平 | |
| **完 成 时 间：** | 2019年1月16日 | |

1. **概述**

随着互联网的快速发展，各类社交媒体平台与日俱增，而微博更是集成了传统网站、论坛、博客等的优点，加上人之间的互动性、关系亲密程度等多种智能算法，并以简练的形式让数据爆发性的传播，促进了人与人之间的交流。网民可以通过微博来分享自己的生活，同时抒发自己的喜怒哀乐。因此对微博每天产生的信息量的分析和利用的需求显得更为迫切。

情感分析，也称倾向性分析、意见抽取和意见挖掘。主要是通过对带有情感色彩的主观性文本进行分析、处理、归纳然后进行推理的过程。而微博人口基数大，涉及的话题广泛，对人们的日常生活产生了不可估量的影响，而对微博的情感分析，更是有着十分重要的意义。

本项目通过使用多种不同算法——SVM,朴素贝叶斯算法以及AdaBoost算法，对微博数据进行情感分析，主要分为中性，乐观和悲观3种类别。通过对于微博评论进行情感分析，旨在了解用户情绪状态，进而发现数据中的价值。通过可视化界面可以训练和测试贝叶斯、SVM、Adaboost算法的模型并选取训练数据的一部分作为测试数据，除此之外还可以选择其他测试文件或者输入文本进行情感分析，并给出简单的分析结果。

本次项目的小组分工如下：

柯楚雯：数据的查找与下载，PPT制作，本论文第二章第一小节与第四章的编写；

余维航：项目代码的编写，代码说明文档与本论文第二章第二小节、第五章的编写；

郑泉斌：数据清洗，本论文第一、三章的编写。

1. **数据获取与预处理**
2. **数据获取**

系统所使用的数据集来源：中国爬盟网站<http://www.cnpameng.com/>，利用爬虫软件爬取热点话题和普通发布的内容作为训练和测试数据。

1. **数据预处理**

为了对微博内容进行情感分类，需要对数据进行预处理，将文本内容转换成相应的词向量，这样有利于使用各种算法进行训练和分类。本项目使用的方法是

首先对微博内容进行分词和清洗。由于微博内容大部分是第一人称视角描述，因此会有很多语气词、连接词，除此之外还有一些微博含有网址链接、用户名等没有情感代表的干扰项，因此把这些干扰语句情感判断的情况都归为停留词。通过分词和清洗，可以去除微博内容中的停留词。利用jieba将文本内容分词，在清洗的时候读取停留词文件，停留词来源于提前准备的文本文件，并将这些词从分词内容中删除。

其次统计清洗后的分词结果中各个词汇出现的频数，取频数的前20%作为关键词，从而根据特征词构成文本的词向量。根据特征词的个数n构建一个大小为1n的一维数组keyword\_list，keyword\_list的每一列代表该特征词在微博文本中出现的次数，如果该文本中有某一个特征词，则在那个特征词对应的keyword\_list列加一，最终形成一条词向量。

因此根据该思想，对文本内容的处理流程如下：

1. 选取特征

选取特征就是选出分类对象中所展现的部分特点，用来实现分类的依据。例如：能拿奖品这种事一定要和大家说一说的，否则以后还怎么在一起愉快的玩耍[酷]我正在领取#微博等级专享礼#，这里的奖品太给力了，快去试试人品吧http://t.cn/RLAxtki。

特征是从大量的文本中选举出来的高频词，假设样本中有许多类似这样的句子，那么由上面该句子我们可以知道，“奖品”、“微博等级”是该句子的高频词，也是影响其分类的主要特征。

1. 降噪

在上面这个例子中，经过降噪处理后：['拿', '这种', '否则', '愉快', '我', '说', '力', '以后', '大家', '太', '博', '一起', '[酷]', '在', '还', '奖品', '了', '给', '事', '吧', '礼', '#', '这里', '能', '等级', '专', '要', '人品', '试试', '怎么', '正', '微', '享', '的', '领取', '一', '和', '去', '，', '快', '一定', '玩耍']

1. 特征降维

特征降维,即降低特征的维数，降低复杂度。在自然语言处理中，多余的特征有时候会影响或误导学习器，特征太多就需要调整更多的参数，过拟合的风险也就越大。本文中，主要是把无关或冗余的特征删掉，特征数量减少了之后不经能够可以加快算法计算的速度，还可以减少干扰，提高分类的准确率。在自然语言处理中，主要是通过一定的统计方法来找出信息量丰富的特征。其中，这些方法包括：信息熵、卡方统计、词频、文档频率、互信息等。

在本项目中使用的是词频统计，也就是选在语料库中出现频率高的词。比如我可以选择语料库中词频最高的n个词（降噪后）作为特征。之后，再将所有的原始文本转化为特征表示的形式。

1. 将文本样本变成特征显示

所以，上述文本经过选取特征之后如下：[{'试试': 'True', '正': 'True', '礼': 'True', '人品': 'True', '专': 'True', '力': 'True', '事': 'True', '玩耍': 'True', '说': 'True', '博': 'True', '微': 'True', '[酷]': 'True', '享': 'True', '奖品': 'True', '领取': 'True', '太': 'True', '等级': 'True', '愉快': 'True'}, 'adv']

1. 将文本随机分成训练集、测试集

本文使用python中的random里的shuffle来将数据随机化，之后再将文本分为训练集和测试集，除此之外还爬取了样本数量相对较少的文本内容作为测试数据。

1. **算法介绍**
2. **朴素贝叶斯**

贝叶斯分类算法是统计学的一种分类方法，它是一类利用概率统计知识进行分类的算法。在许多场合，朴素贝叶斯分类算法可以与决策树和神经网络分类算法相媲美，该算法能运用到大型数据库中，而且方法简单、分类准确率高、速度快。

根据贝叶斯定理：，其是由 P(Y,X)=P(Y|X)P(X)=P(X|Y)P(Y) 推导而来。其中P(Y)为先验概率，叫做后验概率，P(Y,X)一般称作联合概率。

标准的朴素贝叶斯分类算法描述如下：

* 对样本进行人工标记
* 对不同类别的样本进行中文分词
* 去除样本中的垃圾词条
* 将整理后的词条做成特征组，分析并计算词条频率信息
* 根据词条的信息，计算其先验概率
* 读取训练的样本，进行分词，降噪，然后形成样本特征组
* 将测试样本的先验概率代入朴素贝叶斯公式并计算后验概率，得到最大概率的所属类别就是文本的类别。

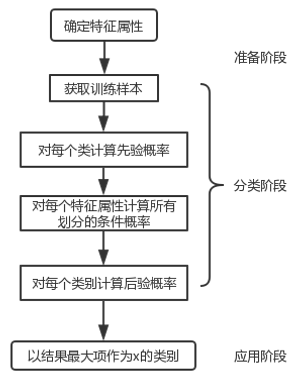
由于贝叶斯定理假设一个属性值对给定类的影响独立于其它属性的值，而此假设在实际情况中经常是不成立的，因此其分类准确率可能会下降。

设每个数据样本用一个n维特征向量来描述n个属性的值，即：X={x1，x2，…，xn}，假定有m个类，分别用C1…Cm表示。给定一个未知的数据样本X（即没有类标号），若朴素贝叶斯分类法将未知的样本X分配给类Ci，则一定是P(Ci|X)>P(Cj|X) 1≤j≤m，j≠i。

根据贝叶斯定理，由于P(X)对于所有类为常数，最大化后验概率P(Ci|X)可转化为最大化先验概率P(X|Ci)P(Ci)。如果训练数据集有许多属性和元组，计算P(X|Ci)的开销可能非常大，为此，通常假设各属性的取值互相独立，这样先验概率P(x1|Ci)，P(x2|Ci)，…，P(xn|Ci)可以从训练数据集求得。

根据此方法，对一个未知类别的样本X，可以先分别计算出X属于每一个类别Ci的概率P(X|Ci)P(Ci)，然后选择其中概率最大的类别作为其类别。

朴素贝叶斯算法成立的前提是各属性之间互相独立。当数据集满足这种独立性假设时,分类的准确度较高，否则可能较低。另外，该算法没有分类规则输出。其流程主要如下图所示：



第一阶段，准备工作阶段。此阶段主要是对文本进行预处理，先对样本进行标注，之后根据词频筛选部分的特征词。该阶段输入的事所有待分类的样本，然后得出特征属性和训练样本。朴素贝叶斯的分类器的准确性主要由筛选出来的特征属性来决定。

第二阶段，分类器训练阶段。根据样本中的频率，然后由每个特征计算出每个类别的先验概率。此阶段主要是根据公式的机械计算。此阶段是朴素贝叶斯分类最重要的一个环节。

第三阶段，应用阶段。该阶段主要将测试样本进行输入，然后由分类器来计算出分类的记过。

对于文本分类，常用的朴素贝叶斯主要存在三种不同的贝叶斯模型：高斯模型、多变量的伯努利模型和多项式模型。高斯模型假设特征是正态分布的，它的一般使用场景是，给出人物的宽度和高度，判断这个人的性别，而情感分析从给定推文文本中提取出词语的个数，不适合正态分布。伯努利模型和高斯模型相似，更适于判断词语是否出现二值特征，而不是词频统计。多项式模型假设特征就是出现次数。这和我们是相关的，因为我们会把推文中的词频当做特征。

该多项式模型中，文本的表示是欧式空间中的一个带有单词频率信息的向量，当一个词语在一篇文章中出现时，该文本的对应的词语频率为1，否则为0。

在条件独立的情况下，给定测试文档d=<,,…,>,多项式朴素贝叶斯根据下面的极大后验概率来对文本进行分类：

具体算法实现过程描述如下：

1. 训练集TrainingSet={(,), (,)…(,)}包含N条训练数据，其中， T是M维向量，属于K类中的一类。
2. 首先，我们来计算先验概率p(y=)

其中I(x)为指示函数，若括号内成立，则计1，否则为0。

1. 接下来计算分子中的条件概率，设M维特征的第j维有L个取值，则某维特征的某个取值，在给定某分类下的条件概率为：

通过学到的概率，给定未分类新实例X，就可以通过上述概率进行计算，得到该实例属于各类的后验概率p(y=|X)，因为对所有的类来说，分母的值都相同，所以只计算分子部分即可，具体步骤如下：

1. 计算该实例属于类的概率：
2. 确定该实例所属的分类y：

其中最大的便是我们的分类。

1. **SVM**

支持向量机（support vector machine，SVM）不仅用于模式分类还可以用于非线性回归，主要做法是在样本点所在的向量空间中，找出一个使分类要求的最优分类超平面，该平面能把不同类的样本分开，不仅能够满足分类精通的要求，还能使两侧的空白区域（分类间隔）最大化。该分类算法具有坚实的统计学理论基础，而且还在许多实际应用比如手写数字的识别、文本分类等都展示了特别明显的实践效用，是一种已经越来越被研究者关注的分类算法。

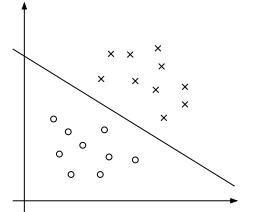
SVM是通过一个非线性映射p，把样本空间映射到一个高维乃至无穷维的特征空间中，使得在原来的样本空间中非线性可分的问题转化为在特征空间中的线性可分的问题。常用的SVM主要依赖于不同的核函数，常用的几种核函数如下：

1. 高斯径向基核：；
2. 多项式核：
3. B-样条核：；
4. 傅里叶核：；

一般使用X表示输入空间，Y表示输出域。通常X∈Rn，对于两类问题，Y = {-1, 1};对于多类问题，Y = {1,2,…,m}; 训练集是训练数据（样本）的集合，通常S=((x1,y1),..(xn, yn))∈（X×Y）n，其中n是样本数目，xi是样本(向量)，yi是它们的标记。

在一维空间里线性函数就是一个点，在二维空间里就是一条直线，三维空间里就是一个平面，如果不关注空间的维数，这种线性函数还有一个统一的名称——超平面。

如下图所示，一个以圆点和“x”的点分布在二维平面上，用一条直线的线把圆点和“x”的点分开来。这条直线的线也就是我们所说的超平面，假设超平面两边的数据点对应的是-1和1。接着，我们可以令分类函数：。显然，如果，那么x是位于超平面上的点。我们不妨要求对于所有满足的点，其对应的y等于-1 ，而则对应y=1的数据点。



然而现实情况下，线性可分问题仅占少数，甚至不存在，我们遇到的所要求取的问题基本上都是非线性的。此时，我们就可以采用支持线性向量分类机。线性支持向量分类器一般用在线性不可分的数据集，即将数据集当做近似可分。我们也可以通过求解得出对应的凸二次规划问题，同时求出超平面。

直观上，我们假设存在一个最优的超平面，其公式为：

对于所有的点集x，都存在能够平行于最优超平面的边界面。即或者。如果要对图3.2中的进行划分，则需要放弃间隔值取正。本文中引入松弛条件，i=1,2,…,l，此时，问题转化为：。将此公式写成等价形式：

其中C>0为惩罚函数，利用拉格朗日变换，函数变为：

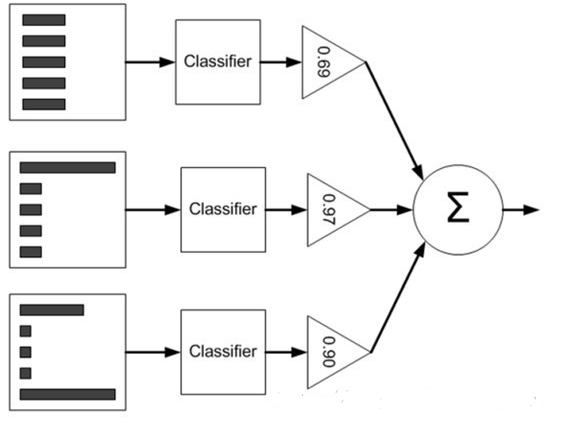
对，求导置0后代入原公式，得到起对偶：

对于LinearSVC的分类问题，就是寻找使得拉格朗日函数L最大，即：

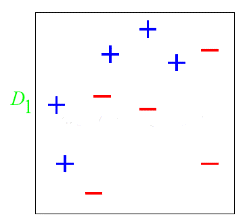
1. **Adaboost**

Adaboost（Adaptive Boosting）也称自适应算法，最早由Freund和Schapine提出。该算法的主要思想是对那些很难分类的模式给予更多的关注。总关注度由一个权值来量化，然后赋予训练集中的样本。

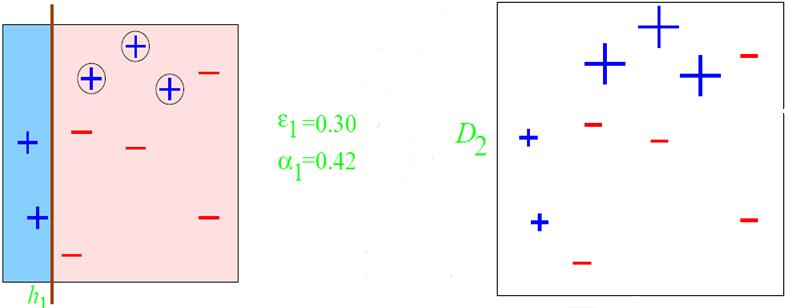
Adaboost是一种迭代的算法，会对同一个训练集使用不同的分类器训练，之后，再把这些分类器集合起来，构建一个最终的最强的分类器。其算法本身是通过改变一个权重D的分布来实现的，该权重D初始化一致，然后改变之后交给下一次分类器。使用Adaboost分类器能够过滤掉一些不必要的训练数据特征，然后放在关键的训练数据上面。AdaBoost分类流程图如下图。



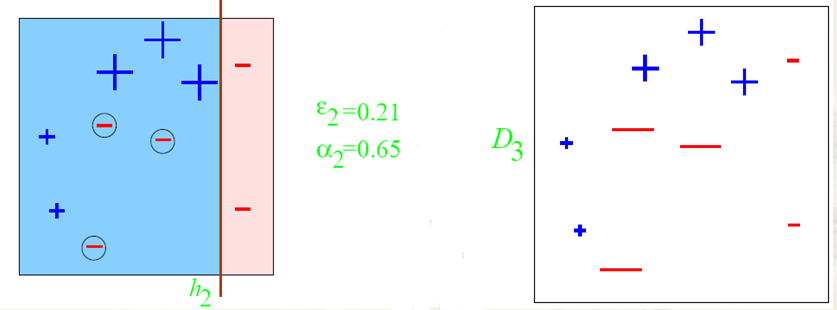
举个例子来说明AdaBoost的实现过程：图中主要有“+”和“-”号，现使用决策树对其进行划分，主要将“+”和“-”号分割开来。



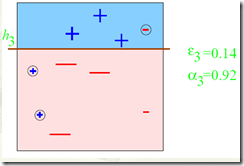
初始分布采用均匀分布即。第一个分类器决策之后的边界如下图。



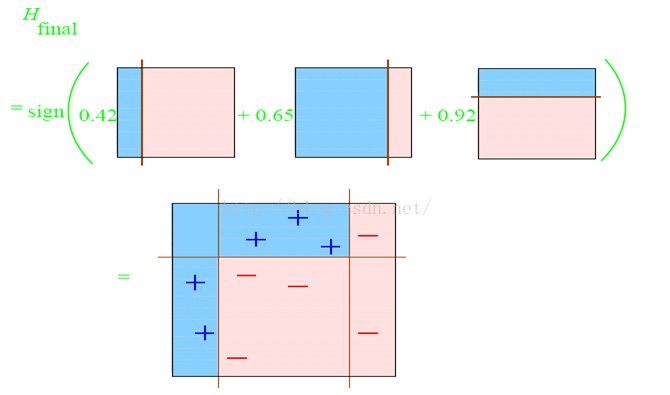
经过第一次分类之后，会得到一个新的样本权重，一个新的子分类器。但是，由图中可以看出，经过第一次划分之后，有三个“+”的被分错误。



进行第二次迭代之后，再次得到一个新的样本权重D3，第二个子分类器h2。由图中可以看出，第二次分类仍是有三个“-”被分类错误。



第三次分类之后，其中有一个“-”和两个“+”被分类错误。再然后我们整合所有的分类器得出结果如下。将三个分类器结合起来即可得出一个最强的分类器。

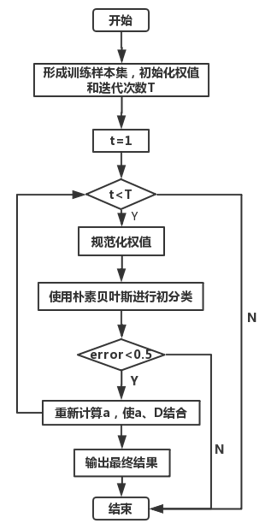


1. **算法具体实现描述**

令训练样本集D={(,), (,)…(,)}，，X用来表示训练样本集空间，是某一类别集。每次迭代的索引为t=1,2，…,T，AdaBoost算法在训练样本上维护一套权重分布W，其中，每个训练样本都对应一个权重，初始时，对所有i都有=1/N。伪代码如下：

|  |
| --- |
| 1. input：   N个训练样本{(,), (,)…(,)}，其中，类标签；  迭代次数T；  弱分类算法weaklearn。 |
| 1. 初始化：赋予每个样本相等的权重：=1/N。 |
| 1. for t=1 to T Do： 2. 在训练样本集D上，利用样本权重和weaklearn学习得到弱分类器：X; 3. 计算弱分类器的错误率：   ，I(  If ：  退出循环   1. 计算： 2. 根据错误率更新样本的权重：   如果分类器结果正确即：if  如果分类器结果错误即：if |
| 1. Output：最终分类器结果可以从中选出 |

1. **算法的流程图**



1. **利用AdaBoost提升朴素贝叶斯文本分类**

将AdaBoost算法直接用于朴素贝叶斯算法的时候，在每次迭代过程中，训练样本如果被错误分类，权重将增加，否则将减少。

文本分类中，文本通常使用空间向量模型，即对于每个文本，都有一个特征向量=()，描述，其中，为该文本中提取出来的的权重。在使用朴素贝叶斯作为AdaBoost的基分类器时，分类过程中我们通常需要计算两个参数：

* 文本属于每个类的先验概率；
* 每个基于每个类的条件概率；

从上一章我们可以知道，朴素贝叶斯公式有：

其中，先验概率

而每个特征基于类的条件概率为：

其中，为特征出现的次数。如果文本属于类别时，，否则等于0。N和L分别代表的是训练文本数和类别数，V为训练文本集的特征集合。

在我们进行文本分类的时候，需要计算某个文本属于每个类别的后验概率，最后将其分到后验概率最大的一个类别。

防止乘积算出来的概率太低接近于0，故计算的时候取对数：

AdaBoost在迭代的时候，会为每个训练样本分配的权重为，然后将其引入参数，上述公式随之会变为：

因此，随着AdaBoost 的每次迭代，样本权重每次都有更新，朴素贝叶斯的先验概率和后验概率都有变化，对朴素贝叶斯分类器的分类产生了扰动，增加了朴素贝叶斯分类器的相异性。

集成方法通过将多个分类器的分类结果组合起来，使最终分类分类器比最初的简单单分类器更好的结果。同时，多分类器能够解决单分类器的一些问题，比如过拟合问题。本文将使用朴素贝叶斯作为AdaBoost的基分类器，然后对微博进行情感分类。由实验可知，AdaBoost确实有效的提升了朴素贝叶斯的分类准确性。

1. **实现**

根据上述三种算法，我们实现了一个文本情感分析系统，可以实现对输入的文本进行情感分析，判断文本的情感极性。本系统对文本情感的判断包括二分类和多分类，其中二分类是将情感分为积极和消极两类，多分类是将情感分为积极、消极和中立三类。对于这两种分类方法，本系统都用了三种算法来进行分析，这三种算法分别是上文所提到的SVM分类算法、贝叶斯算法和AdaBoost算法。图4.1是本项目的系统界面，可以选择文件进行情感分析，也可以手动输入文本进行情感分析。

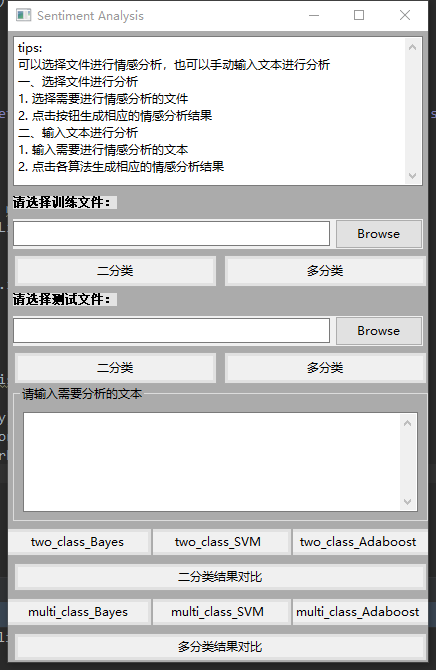


图4.1 系统界面

1. **选择文件测试**

对文本进行情感分析，首先需要上传情感分析所需要的训练文件对模型进行训练。本项目所使用的训练文件来源于中国爬萌网站上的微博数据。

如图4.2所示是上传训练文件并进行二分类的训练。在上传训练文件之后，分别调用三种算法对模型进行二分类训练，并进行了多次测试，计算三种算法的错误率。在第一次测试中，多项式贝叶斯算法的错误率为0.34，SVM算法的错误率为0.28，AdaBoost算法的错误率为0.22。在第二次测试中，多项式贝叶斯算法的错误率为0.37，SVM算法的错误率为0.21，AdaBoost的错误率为0.2。综合两次测试的结果来看，多项式贝叶斯算法的错误率最高，AdaBoost算法的错误率最低。

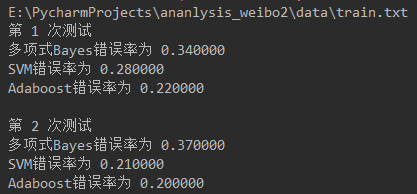


图4.2 二分类测试错误率

如图4.3所示是三种二分类算法的平均错误率和时间，可以看出adaboost算法的平均错误率是最低的，SVM其次，贝叶斯算法的平均错误率最高。而训练所用的时间是贝叶斯最短，SVM其次，adaboost最长。

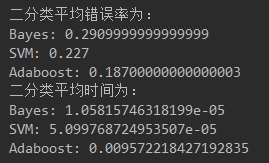


图4.3 二分类平均错误率和时间

综合比较三种算法的错误率，如图4.4所示，可以看出三种算法均在迭代次数为1和10次时平均错误率最大，SVM算法和adaboost算法都在迭代次数为3时取到最小值，而贝叶斯算法在迭代次数为4是取到最小值。从整体来看，贝叶斯的错误率折线在最上方，SVM的折线处于中间，adaboost处于最下方。也就是说，贝叶斯的错误率最大，adaboost的错误率最小。

从表格中可以看出，二分类时三种算法的具体平均错误率和平均运行时间。可以看出对于二分类问题，adaboost算法的平均错误率最低，SVM其次，贝叶斯算法最高。而平均运行时间贝叶斯最短，SVM其次，adaboost最长。

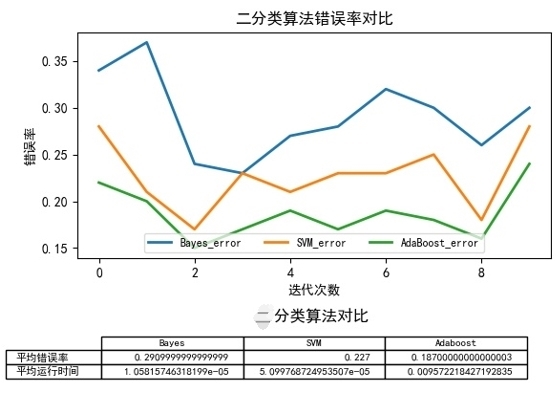


图4.4 二分类算法错误率和运行时间对比

如图4.5所示是上传训练文件并进行多分类的训练。跟二分类训练同样，在上传训练文件之后，分别调用三种算法对模型进行多分类训练并计算三种算法的错误率。可以看出多项式贝叶斯算法的错误率为0.32，SVM算法的错误率为0.13，AdaBoost算法的错误率为0.16。即多项式贝叶斯算法的错误率最高，SVM算法的错误率最低。

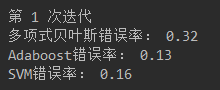


图4.5 多分类测试错误率

如图4.6所示是三种多分类算法的平均错误率和时间，可以看出adaboost算法的平均错误率是最低的，SVM其次，贝叶斯算法的平均错误率最高。而训练所用的时间是贝叶斯最短，SVM其次，adaboost最长。

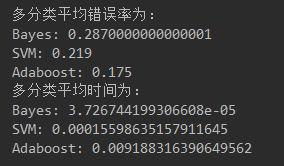


图4.6 多分类平均错误率和时间

综合比较三种算法的错误率，如图3.7所示，可以看出SVM算法和adaboost算法都在迭代次数为1时取到最小值，而贝叶斯算法在迭代次数为9时取到最小值。从整体来看，贝叶斯的错误率折线在最上方，SVM的折线处于中间，adaboost处于最下方。也就是说，贝叶斯的错误率最大，adaboost的错误率最小。

从表格中可以看出，在进行多分类时三种算法的具体平均错误率和平均运行时间。可以看出对于多分类问题，adaboost算法的平均错误率最低，SVM其次，贝叶斯算法最高。而平均运行时间贝叶斯最短，SVM其次，adaboost最长。

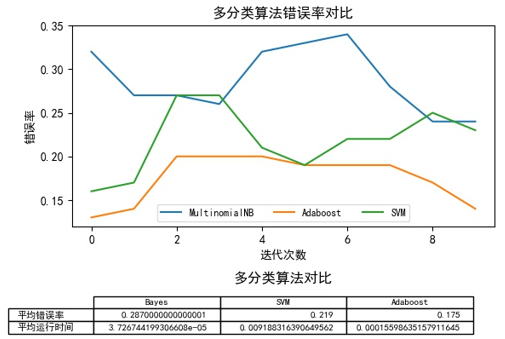


图4.7 多分类算法错误率和运行时间对比

在系统中可以选择文件并上传来进行情感分析的测试。在本次测试中，选用的数据来自于中国爬萌网站中的微博文本数据。

如图3.8所示是对测试文件进行二分类之后得到的错误率，其中多项式贝叶斯的错误率为0.41，SVM的错误率为0.45，adaboost的错误率为0.34。显然，adaboost的效果要优于其余两种算法。而效果最差的算法为SVM算法。

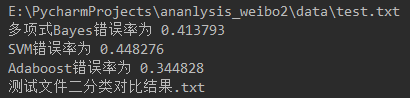


图4.8 二分类测试结果

如图4.9所示是对测试文件进行多分类得到的错误率结果。其中多项式贝叶斯的错误率为0.38，SVM的错误率为0.41，adaboost的错误率为0.28。显然，在多分类的测试中，adaboost的效果也是要优于其余两种算法。而效果最差的算法仍然为SVM算法。

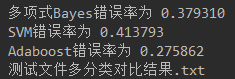


图4.9 多分类测试结果

1. **输入文本测试**

在系统中，还可以再文本框中输入一段文字，并选择算法来对这段文字进行情感极性的判断。如图4.10所示是本次测试所输入的文本。

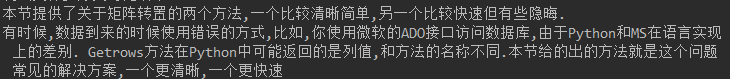


图4.10 输入文本

对该段文本进行二分类，判断这段文本的情感极性。如图4.11所示是使用三种算法对其进行判断所得到的结果。其中贝叶斯算法将这段文本判断为消极，SVM和adaboost算法将这段文本的情感判断为消极。

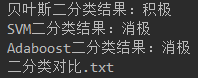


图4.11 二分类结果

然后对该段文本进行多分类，再次判断这段文本的情感极性。如图4.12所示是使用三种算法对其进行判断所得到的结果。可以看到贝叶斯算法将这段文本判断为消极，SVM和adaboost算法将这段文本的情感判断为消极。使用多分类对这段文本判断的结果与二分类的结果一致。

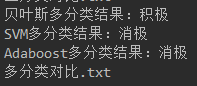


图4.12 多分类结果

1. **总结与展望**
2. **本文主要内容总结**

本文按照传统的文本分析方法，并根据微博的特点进行了情感分析。主要在以下方面展开了研究：

1. 对微博的降噪清理，对无关性的微博本文进行过滤。从微博短文本的的特点，对文本中的停留词、用户名、网址链接等无关干扰项进行了过滤和删除。
2. 利用多种测试数据，通过可视化界面完成了贝叶斯、SVM、Adaboost在二分类和多分类的性能对比，并给出了相关对比结果。
3. 通过对比结果可以看出，无论是在进行二分类还是多分类时，AdaBoost算法的分类错误率都要低于贝叶斯和SVM算法，这是因为在本项目中使用AdaBoost算法对朴素贝叶斯进行加强，将朴素贝叶斯作为AdaBoost的弱分类器，然后对其进行加强，从实验结果可以看出该方法有效的提高了情感分类的准确性。但同时也能在时间对比数据上可以看出，Adaboost的运行时间要比其他两者略高，这是因为在选择基分类器个数时选取了较大的数字（500），当基分类器个数下降时运行时间也会降低，但同时分类的准确率也会下降。
4. **存在问题及未来研究展望**
5. 本项目在获取训练数据集外的数据内容时采用人工清洗和筛选的方式，工作量较大。
6. 本项目通过特征词库的建立文本的词向量，从原理上看比较符合分类的思想，但是并不是所有的微博特征都十分明显，当训练数据中包含大量的个人情感表达时，整体的特征就会非常不明显，对特征词库的建立就有一定的影响。本项目通过统计所有次出现的频次，取前百分之二十作为特征词库内容建立词向量，这种方法比较适用于特征较为明显的数据，当训练数据非常多且没有明显特征时，训练出的模型泛化效果则不是很理想。同时当训练数据非常大时，特征词库随之增大，则计算词向量的工作将变得比较庞大。