## 朴素贝叶斯分类器应用

学号: 18401190103 姓名： 谢威

学号: 18401190202 姓名： 陆镇涛

学号: 18401190120 姓名： 曹鹏霄

学号: 18401190107 姓名： 王云浩 提交时间： 2020-11-4

## 一、实验目的

1．了解贝叶斯原理；

2．理解朴素贝叶斯算法原理；

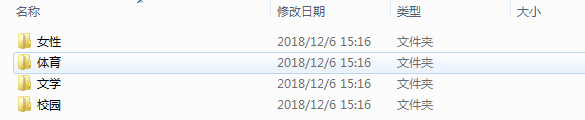
3．掌握朴素贝叶斯工作流程

4．初步应用朴素贝叶斯分类器在分类中应用；

## 实验内容

文档分类

给定4种中文文档类型:女性、体育、文学、校园，其中训练集放在train文件夹里，测试数据放在test文件夹里，停用词放在stop文件夹里。





请使用朴素贝叶斯分类对训练集进行训练，并对测试集进行验证，，并给出测试集的准确率。

## 实现过程分析

1. 使用jieba包对文档进行分词，jieba包中包含中文的停用词表和分词方法

*def get\_data(base\_path, labels): # 获取数据集*

*contents = []*

*# 数据在文件夹下，需要遍历各个文档类别下的文件*

*for label in labels:*

*files = {fileName for fileName in os.listdir(base\_path + label)}*

*for fileName in files:*

*file = open(base\_path + label + '/' + fileName, encoding='gb18030*

*word = jieba.cut(file.read()) # 对文档进行分词*

*contents.append(' '.join(word)) # 用 ' ' 分隔*

*return contents*

1. 加载停用词表需要我们先下载常用的停用词表，然后放到stop\_word.txt中，读取里面的数据放到数组中

*stop\_words = [line.strip() for line in open(stop\_words\_path, encoding='utf-8-sig').readlines()]*

1. 计算单词权重用sklearn里的TfidfVectorizer类，使用fit\_transform方法进行拟合，得到TF-IDF特征空间features，可以理解为选出来的分词就是特征，计算这些特征在文档上的特征向量，得到特征空间features

*tf = TfidfVectorizer(stop\_words=stop\_words, max\_df=0.5) #0.5表示将在一半文档都出现的词汇剔除*

*train\_features = tf.fit\_transform(train\_contents)*

1. 生成朴素贝叶斯分类器把训练集训练出来的特征空间features和训练集对应的分类labels传给贝叶斯分类器，会自动生成一个符合特征空间和对应分类的分类器

*train\_labels = ['体育'] \* 1337 + ['女性'] \* 954 + ['文学'] \* 766 + ['校园'] \* 249*

*clf = MultinomialNB(alpha=0.001).fit(train\_features, train\_labels)*

1. 使用分类器做预测。主要是得到测试集的特征矩阵，先创建FfidfVectorizer，使是和训练集一样的stop\_words和max\_df，然后用FfidfVectorizer类对测试集内容进行拟合，得到测试集特征矩阵，再用训练好的分类器对测试特征矩阵进行预测，predict这个函数求解所有的后验概率并找到最大的那个

*test\_tf = TfidfVectorizer(stop\_words=stop\_words, max\_df=0.5, vocabulary=tf.vocabulary\_)*

*test\_features = test\_tf.fit\_transform(test\_contents)*

*predicted\_labels = clf.predict(test\_features)*

1. 计算准确率主要是对模型预测的结果和实际的结果进行对比，给出模型的准确率，用到sklearn中的metrics包中的accuracy\_score函数

*test\_labels = ['体育'] \* 115 + ['女性'] \* 38 + ['文学'] \* 31 + ['校园'] \* 16*

*print('准确率', metrics.accuracy\_score(test\_labels, predicted\_labels))*

## 源代码

import os

import jieba

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn import metrics

stop\_words\_path = 'stop/stopword.txt'

train\_base\_path = 'train/'

test\_base\_path = 'test/'

train\_labels = ['女性', '体育', '文学', '校园']

test\_labels = ['女性', '体育', '文学', '校园']

def get\_data(base\_path, labels): # 获取数据集

contents = [] for label in labels:

files = {fileName for fileName in os.listdir(base\_path + label)}

for fileName in files:

file = open(base\_path + label + '/' + fileName, encoding='gb18030')

word = jieba.cut(file.read())

contents.append(' '.join(word)) # 因为切词，所以用 ' ' 分隔

print(fileName + '文件读取错误')

return contents

train\_contents = get\_data(train\_base\_path, train\_labels)

test\_contents = get\_data(test\_base\_path, test\_labels)

stop\_words = [line.strip() for line in open(stop\_words\_path, encoding='utf-8-sig').readlines()]

tf = TfidfVectorizer(stop\_words=stop\_words, max\_df=0.5)

train\_features = tf.fit\_transform(train\_contents) # 拟合

train\_labels = ['体育'] \* 1337 + ['女性'] \* 954 + ['文学'] \* 766 + ['校园'] \* 249

clf = MultinomialNB(alpha=0.001).fit(train\_features, train\_labels)

test\_tf = TfidfVectorizer(stop\_words=stop\_words, max\_df=0.5, vocabulary=tf.vocabulary\_)

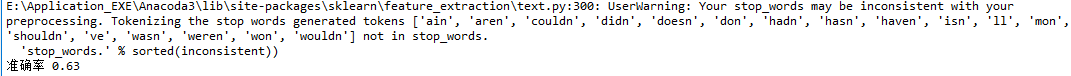
test\_features = test\_tf.fit\_transform(test\_contents)

predicted\_labels = clf.predict(test\_features)

test\_labels = ['体育'] \* 115 + ['女性'] \* 38 + ['文学'] \* 31 + ['校园'] \* 16

print('准确率', metrics.accuracy\_score(test\_labels, predicted\_labels))

1. **实验结果**



1. **结论**

对女性、体育、文学、校园四种类型的文档用朴素贝叶斯分类对训练集训练，并对测试集进行验证，最后的准确率为0.63