AIL2-2017Z8009061078-李中欢-人工智能概论课程实验报告-Bayes

Python

1.实验目的

（1）理解朴素贝叶斯算法的基本原理
（2）学会使用TensorFlow编写基本的贝叶斯程序
（3）使用朴素贝叶斯使用Python进行文本分类
（4）加强Python语言的了解
（5）学会使用Python开发环境

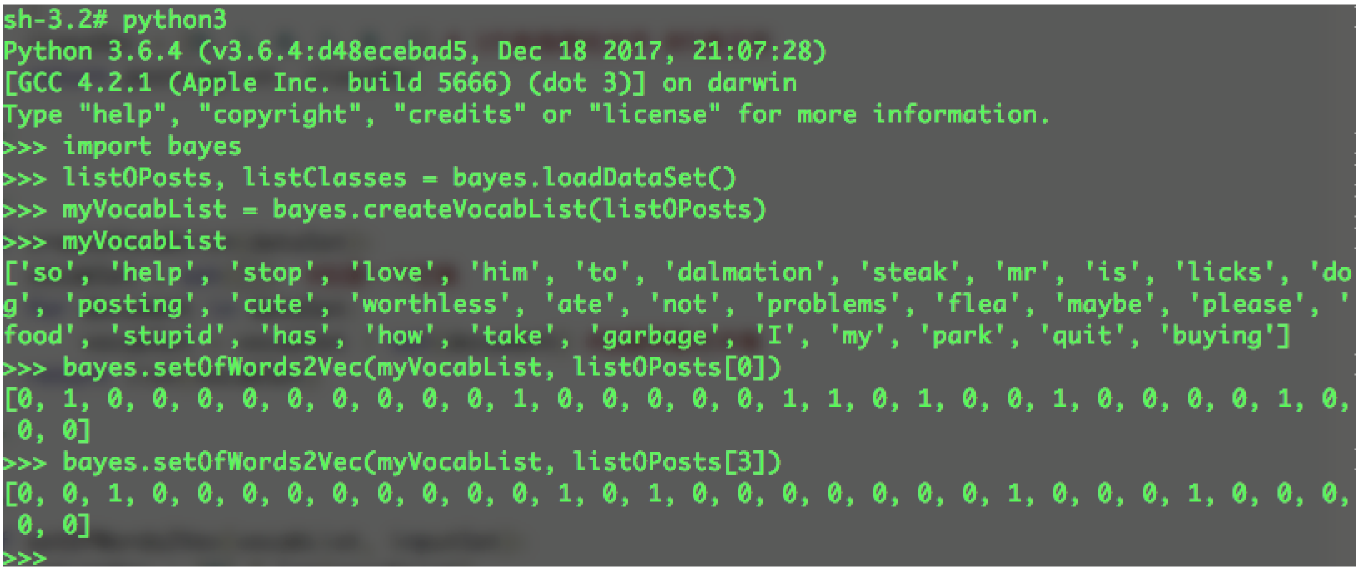
2.实验准备

（1）下列组件是完成本实验所必须的
TensorFlow安装包；
Python 3.6.x；
（2）实验用数据；

3.实验内容和步骤

1. 从文本中创建词向量bayes.py

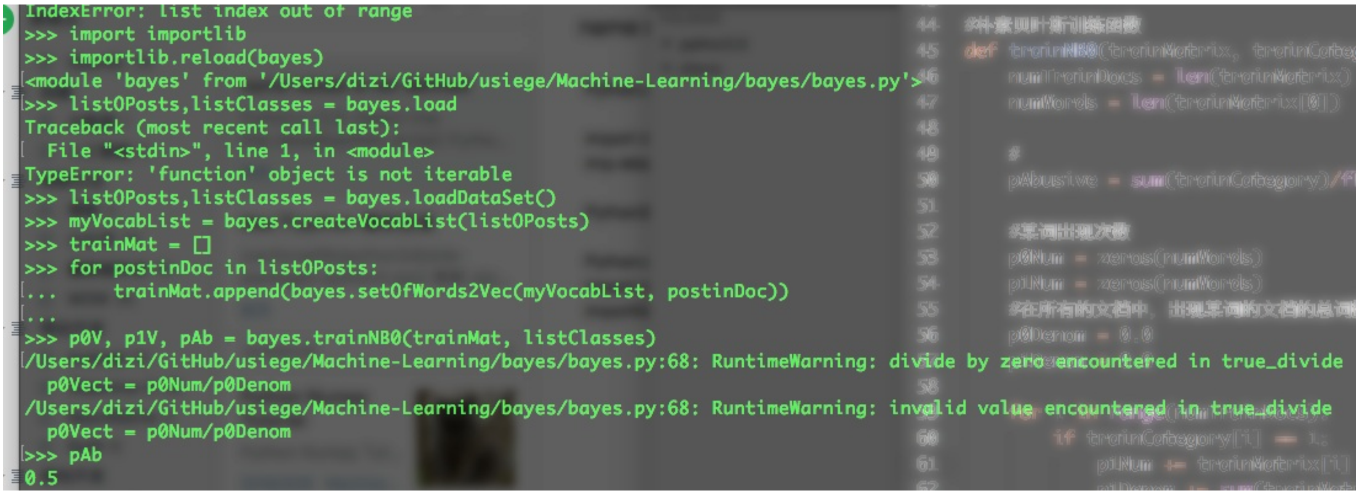
```
#!usr/bin/python
#-*-encoding:utf-8-*-

'''
该函数返回实验样本，该样本被切分成词条集合；
第二个变量返回类别，该类别由人工标注，用于训练程序以便自动检查侮辱性留言；
'''
def loadDataSet():
    postingList = [
        ['my','dog','has','flea','problems','help','please'],
        ['maybe','not','take','him','to','dog','park','stupid'],
        ['my','dalmation','is','so','cute','I','love','him'],
        ['stop','posting','stupid','worthless','garbage'],
        ['mr','licks','ate','my','steak','how','to','stop','him'],
        ['quit','buying','worthless','dog','food','stupid']
    ]
    classVec = [0, 1, 0, 1, 0, 1] # 1代表侮辱性文字 0代表正常
    return postingList, classVec

'''
'''
def createVocabList(dataSet):
    vocabSet = set([])  #创建一个空集
    for document in dataSet:
        vocabSet = vocabSet | set(document) #创建两集合并集
    return list(vocabSet)

'''
该函数输入参数为词汇表及某个文档，输出的是文档向量，向量每一元素为1or0，分别表示词汇表中的单词在输入文档中是否出现
'''
def setOfWords2Vec(vocabList, inputSet):
    returnVec = [0] * len(vocabList)
    for word in inputSet:
        if word in vocabList:
            returnVec[vocabList.index(word)] = 1
        else:
            print("the word: %s is not in my Vocabulary!" % word)
    return returnVec
```



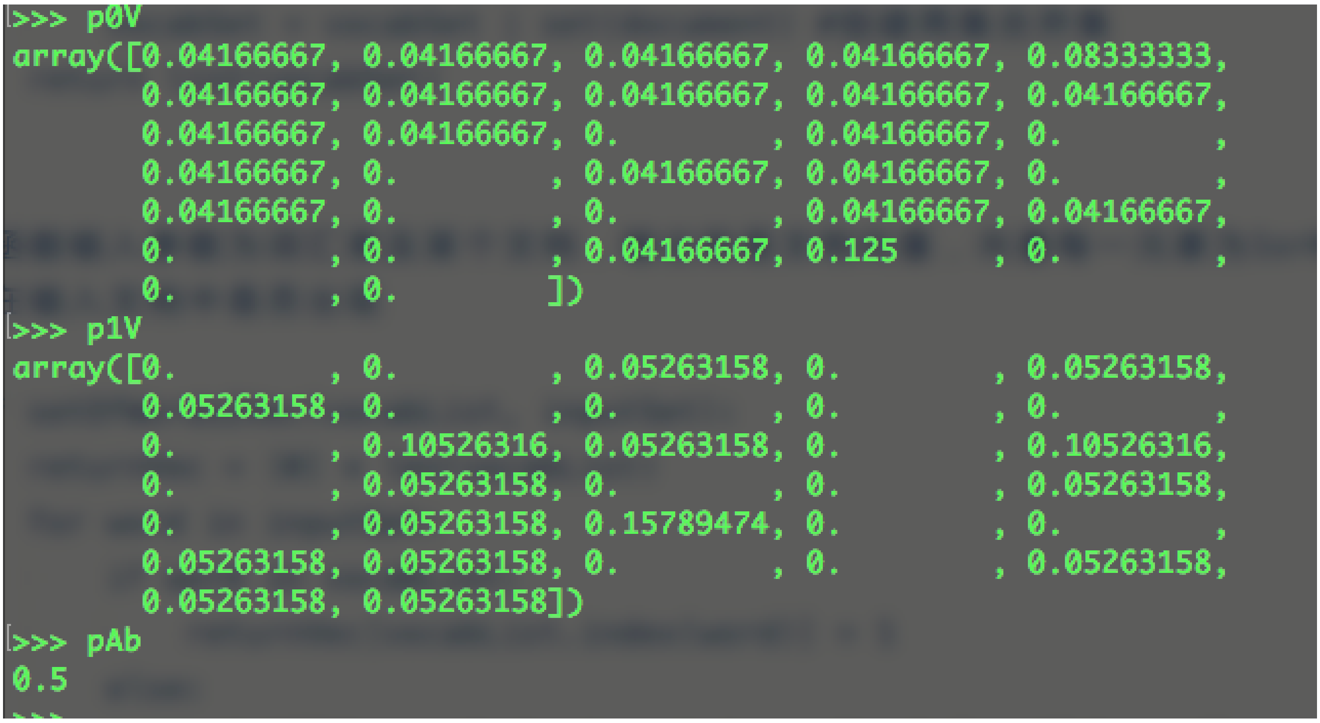
```
sh-3.2# python3
Python 3.6.4 (v3.6.4:d48ecebad5, Dec 18 2017, 21:07:28)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import bayes
>>> listOPosts, listClasses = bayes.loadDataSet()
>>> myVocabList = bayes.createVocabList(listOPosts)
>>> myVocabList
['so', 'help', 'stop', 'love', 'him', 'to', 'dalmation', 'steak', 'mr', 'is', 'licks', 'do
g', 'posting', 'cute', 'worthless', 'ate', 'not', 'problems', 'flea', 'maybe', 'please', '
food', 'stupid', 'has', 'how', 'take', 'garbage', 'I', 'my', 'park', 'quit', 'buying']
>>> bayes.setOfWords2Vec(myVocabList, listOPosts[0])
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
 0, 0]
>>> bayes.setOfWords2Vec(myVocabList, listOPosts[3])
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
 0, 0]
>>>
```

2. 朴素贝叶斯训练函数

```
#朴素贝叶斯训练函数
def trainNB0(trainMatrix, trainCategory):
    numTrainDocs = len(trainMatrix)
    numWords = len(trainMatrix[0])

    #
    pAbusive = sum(trainCategory)/float(numTrainDocs)

    #某词出现次数
    p0Num = zeros(numWords)
    p1Num = zeros(numWords)
    #在所有的文档中，出现某词的文档的总词数
    p0Denom = 0.0
    p1Denom = 0.0

    for i in range(numTrainDocs):
        if trainCategory[i] == 1:
            p1Num += trainMatrix[i]
            p1Denom += sum(trainMatrix[i])
        else:
            p0Num += trainMatrix[i]
            p0Denom += sum(trainMatrix[i])

    p1Vect = p1Num/p1Denom
    p0Vect = p0Num/p0Denom

    return p0Vect, p1Vect, pAbusive
```



```
IndexError: list index out of range
>>> import importlib
>>> importlib.reload(bayes)
<module 'bayes' from '/Users/dizi/GitHub/usiege/Machine-Learning/bayes/bayes.py'>
>>> listOPosts,listClasses = bayes.load
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'function' object is not iterable
>>> listOPosts,listClasses = bayes.loadDataSet()
>>> myVocabList = bayes.createVocabList(listOPosts)
>>> trainMat = []
>>> for postinDoc in listOPosts:
...     trainMat.append(bayes.setOfWords2Vec(myVocabList, postinDoc))
...
>>> p0V, p1V, pAb = bayes.trainNB0(trainMat, listClasses)
/Users/dizi/GitHub/usiege/Machine-Learning/bayes/bayes.py:68: RuntimeWarning: divide by zero encountered in true_divide
  p0Vect = p0Num/p0Denom
/Users/dizi/GitHub/usiege/Machine-Learning/bayes/bayes.py:68: RuntimeWarning: invalid value encountered in true_divide
  p0Vect = p0Num/p0Denom
>>> pAb
0.5
```



```
>>> p0V
array([0.04166667, 0.04166667, 0.04166667, 0.04166667, 0.08333333,
       0.04166667, 0.04166667, 0.04166667, 0.04166667, 0.04166667,
       0.04166667, 0.04166667, 0.        , 0.04166667, 0.        ,
       0.04166667, 0.        , 0.04166667, 0.04166667, 0.        ,
       0.04166667, 0.        , 0.        , 0.04166667, 0.04166667,
       0.        , 0.        , 0.04166667, 0.125     , 0.        ,
       0.        , 0.        ])
>>> p1V
array([0.        , 0.        , 0.05263158, 0.        , 0.05263158,
       0.05263158, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.10526316, 0.05263158, 0.        , 0.10526316,
       0.        , 0.05263158, 0.        , 0.        , 0.05263158,
       0.        , 0.05263158, 0.15789474, 0.        , 0.        ,
       0.05263158, 0.05263158, 0.        , 0.        , 0.05263158,
       0.05263158, 0.05263158])
>>> pAb
0.5
```

3. 修改分类器

* Problem1:计算多个概率的乘积以获得文档属于某个类别概率，如果其中有一个概率值为0，那最后乘积也为0；为降低这种影响，可以将所有词出现初始化为1，并将分母初始化为2

```
        p0Num = ones(numWords);
        p1Num = ones(numWords)
        p0Denom = 2.0;
        p1Denom = 2.0
```

- Problem2: 下溢出，太多很小的数相乘会造成下溢出，解决办法是取自然对数，把乘法转换成加法，通过求对数避免下溢出或者浮点数舍入导致错误

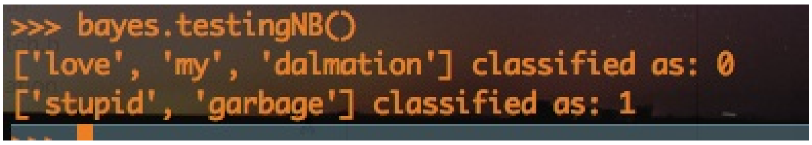
```
    p1Vect = log(p1Num/p1Denom)
    p0Vect = log(p0Num/p0Denom)
```

以上；

4. 分类器编写

```
#构建朴素贝叶斯分类函数
def classityNB(vec2Classify, p0Vec, p1Vec, pClass1):
    p1 = sum(vec2Classify * p1Vec) + log(pClass1)
    p0 = sum(vec2Classify * p0Vec) + log(1.0 - pClass1)
    if p1 > p0:
        return 1;
    else:
        return 0;

def testingNB():
    listOPosts, listClasses = loadDataSet()
    myVocabList = createVocabList(listOPosts)
    trainMat = []
    for postinDoc in listOPosts:
        trainMat.append(setOfWords2Vec(myVocabList, postinDoc))
    p0V, p1V, pAb = trainNB0(array(trainMat), array(listClasses))

    testEntry = ['love', 'my', 'dalmation']
    thisDoc = array(setOfWords2Vec(myVocabList, testEntry))
    print(testEntry, 'classified as:', classityNB(thisDoc, p0V, p1V, pAb))

    testEntry = ['stupid', 'garbage']
    thisDoc = array(setOfWords2Vec(myVocabList, testEntry))
    print(testEntry, 'classified as:', classityNB(thisDoc, p0V, p1V, pAb))
```

通过训练器分类得出结果：



```
>>> bayes.testingNB()
['love', 'my', 'dalmation'] classified as: 0
['stupid', 'garbage'] classified as: 1
```

5. 文档词袋模型

```
#文档词袋模型
def bagofWords2VecMN(vocabList, inputSet):
    returnVec = [0] * len(vocabList)
    for word in inputSet:
        if word in vocabList:
            returnVec[vocabList.index(word)] += 1
    return returnVec
```

4. 实验结果及结论

（1）完成情况

使用贝叶斯方法完成了对文档词的分类；

实验结果满足预期结果输出；

……

（2）实验结论

使用贝叶斯对在线社区留言板进行分析，通过对留言内容进行过滤，识别出侮辱类和非侮辱类言论；

……

（3）问题分析

朴素贝叶斯在其他方面的应用？