Lab 4: Linear Algebra and Probability

Pin Qian

March 29, 2021

Module: INT 104
Supervisor: Xiaobo Jin
Xi'an Jiaotong-Liverpool University

Contents

1 Introduction

Linear Algebra and Probability is the foundation of artificial intelligence. In this work, a two-dimensional array is given. Each row of the array has six columns, the first five columns are the feature values and the final column is the label. The task is to compute the prior probabilities and conditional probabilities with python.

2 Design and Implementation

2.1 Design principals

For **Q1**, read the .csv file and parse its contents into a list named *data* using python's own method *list*(). Then, use *np.mat*() method to contvert the list into matrix named *data_mat*. However, since the task dose not need much arithmetical opertation, so matrix oparation is not necessary here to shorten computing time. Thus, in later work, *data* will be used.

For **Q2**, a function *getPriorProb*() is defined to compute the prior probability of each class. The function need two inputs:

Input	Meaning
data	the matrix we got in Q1
lable	the class label which we want to compute the prior probability

In the function, *cnt* is declared as a counter, which holds the number of valid rows that meet the requirement of $row[5]$ equals to the value of input label during the for loop. Finally, the probability is *cnt* divide the length of data.

For **Q3**, a function *getConProb*() is defined to compute the conditional probabilities. The function need four inputs:

Input	Meaning
data	the matrix we got in Q1
index	the index of a row of the matrix
a_tar	the expected value at the index which we want to compute
label	the class label which we want to compute

In the function, Two counters are declared before the for loop. *l_cnt* records the number of how many rows' label equal to the expected label. *cnt* id records the number of how many rows' label equal to the expected label while the feature value equal to the expected feature value. Finally, the probability is the result of *l_cnt* divided by *cnt*.

2.2 Implementation

```
import csv
import numpy as np
csv_file = open('binary_data.csv')
csv_reader = csv.reader(csv_file, delimiter = ',')
data = list(csv_reader)
data_mat = np.mat(data) # Q1: make it to a matrix

def getPriorProb(data, label): # Q2 function
    cnt = 0                    # a counter
    for row in data:           # traverse the data row by row
        if row[5] == label:    # if label equals to target
            cnt += 1           # cnt adds 1
    return cnt / len(data)     # return the result

def getConProb(data, index, a_tar, label): # Q3 function
    cnt = 0                                # a counter
    l_cnt = 0                              # a label counter
    for row in data:                       # traverse the data
        if row[5] == label:                # if label equals to target
            l_cnt += 1                     # l_cnt adds 1
            if row[index] == a_tar:        # if feature value in label
                cnt += 1                   # equal to target, cnt adds 1
    return cnt / l_cnt                     # return the result
# Q1
print('The shape of the matrix is', data_mat.shape)
# Q2
print('p(l=0) =', getPriorProb(data, '0'))
print('p(l=1) =', getPriorProb(data, '1'))
# Q3
for idx in range(0, 5): # given l=0, the conditional p of a_i=0
    print('p(a_%d=0 | l=0) ='%(idx),getConProb(data,idx,'0','0'))
for idx in range(0, 5): # given l=0, the conditional p of a_i=1
    print('p(a_%d=1 | l=0) ='%(idx),getConProb(data,idx,'1','0'))
for idx in range(0, 5): # given l=1, the conditional p of a_i=0
    print('p(a_%d=0 | l=1) ='%(idx),getConProb(data,idx,'0','1'))
for idx in range(0, 5): # given l=1, the conditional p of a_i=1
    print('p(a_%d=1 | l=1) ='%(idx),getConProb(data,idx,'1','1'))
```

3 Test

The results are printed as follows.

```
***************Q1***************
The shape of the matrix: (100, 6)
******Q2*****
p(l=0) = 0.49
p(l=1) = 0.51
******************Q3******************
p(a_0 = 0 | l = 0) = 0.46938775510204084
p(a_1 = 0 | l = 0) = 0.673469387755102
p(a_2 = 0 | l = 0) = 0.46938775510204084
p(a_3 = 0 | l = 0) = 0.4897959183673469
p(a_4 = 0 | l = 0) = 0.40816326530612246
----------------------------------------
p(a_0 = 1 | l = 0) = 0.5306122448979592
p(a_1 = 1 | l = 0) = 0.32653061224489793
p(a_2 = 1 | l = 0) = 0.5306122448979592
p(a_3 = 1 | l = 0) = 0.5102040816326531
p(a_4 = 1 | l = 0) = 0.5918367346938775
----------------------------------------
p(a_0 = 0 | l = 1) = 0.5294117647058824
p(a_1 = 0 | l = 1) = 0.45098039215686275
p(a_2 = 0 | l = 1) = 0.6274509803921569
p(a_3 = 0 | l = 1) = 0.5686274509803921
p(a_4 = 0 | l = 1) = 0.5490196078431373
----------------------------------------
p(a_0 = 1 | l = 1) = 0.47058823529411764
p(a_1 = 1 | l = 1) = 0.5490196078431373
p(a_2 = 1 | l = 1) = 0.37254901960784315
p(a_3 = 1 | l = 1) = 0.43137254901960786
p(a_4 = 1 | l = 1) = 0.45098039215686275
```

4 Conclusion

In conclusion, the prior probability and conditional probability are computed as above. The basic knowledge of probability theory has been consolidated and applied.