Lab 4: Linear Algebra and Probability

Author: Xu Zhao

Module: INT104

Teacher: Xiaobo Jin

Date: 1/4/2021

Introduction

Given a two-dimensional array, where each row represents an instance (or object). For each row, the first 5 columns are the attributes of the instance and the final column is the label of the instance such as a0, a1, a2, a3, a4, l.

In this project, it needs to compute some estimated probabilities

Design & Implementation

Dynamic Programming

1. Design principals

   First, use some given statement to read the file and transform it into a matrix in python:

   Second, using this statement to select the two column ai and l: aix = matrix[:, [x, 5]].

   Third, create four new matrixes which have two columns.

   Finally, start to calculate. The row of the matrix is N: int = len(matrix).

   To calculate the situation when l = 0 and 1, use the statement: l0 = Counter(l)['0'] and l1 = Counter(l)['1']. p(l = 0) = l0/N, p(l=1) = l1/N.

   To calculate p(ai = 0|l = 0), use the statement: p3 = (ai0 == q1).all(1).sum()/l0. The others are the same.

2. Implementation

3.
```
import csv
import numpy as np
from collections import Counter
csv_file = open('binary_data.csv' )
csv_reader = csv.reader(csv_file, delimiter=',')
final_list = list(csv_reader)
matrix = np.array(final_list)


# Read the text file and parse its content into a matrix. (20 scores)
print(matrix)


# Compute the prior probabilities p(l = 0) and p(l = 1) (20 scores)
```

```
#a0 = matrix[:, 0]
#a1 = matrix[:, 1]
#a2 = matrix[:, 2]
#a3 = matrix[:, 3]
#a4 = matrix[:, 4]
l = matrix[:, 5]

ai0 = matrix[:, [0, 5]]
ai1 = matrix[:, [1, 5]]
ai2 = matrix[:, [2, 5]]
ai3 = matrix[:, [3, 5]]
ai4 = matrix[:, [4, 5]]

q1 = np.array(['0', '0'])
q2 = np.array(['1', '0'])
q3 = np.array(['0', '1'])
q4 = np.array(['1', '1'])


N: int = len(matrix)
#print(N)

l0 = Counter(l)['0']
p1 = l0/N
print(p1)

l1 = Counter(l)['1']
p2 = l1/N
print(p2)




#3

##1



p3 = (ai0 == q1).all(1).sum()/l0
print(p3)

p4 = (ai1 == q1).all(1).sum()/l0
```

```
print(p4)

p5 = (ai2 == q1).all(1).sum()/10
print(p5)

p6 = (ai3 == q1).all(1).sum()/10
print(p6)

p7 = (ai4 == q1).all(1).sum()/10
print(p7)


##2
p8 = (ai0 == q2).all(1).sum()/10
print(p8)

p9 = (ai1 == q2).all(1).sum()/10
print(p9)

p10 = (ai2 == q2).all(1).sum()/10
print(p10)

p11 = (ai3 == q2).all(1).sum()/10
print(p11)

p12 = (ai4 == q2).all(1).sum()/10
print(p12)

##3
p13 = (ai0 == q3).all(1).sum()/11
print(p13)

p14 = (ai1 == q3).all(1).sum()/11
print(p14)

p15 = (ai2 == q3).all(1).sum()/11
print(p15)

p16 = (ai3 == q3).all(1).sum()/11
print(p16)

p17 = (ai4 == q3).all(1).sum()/11
print(p17)
```

```
##4
p18 = (ai0 == q4).all(1).sum()/l1
print(p18)

p19 = (ai1 == q4).all(1).sum()/l1
print(p19)

p20 = (ai2 == q4).all(1).sum()/l1
print(p20)

p21 = (ai3 == q4).all(1).sum()/l1
print(p21)

p22 = (ai4 == q4).all(1).sum()/l1
print(p22)
```

Result

The result is:

```
[['1' '1' '1' '0' '1' '1']
 ['1' '0' '0' '0' '1' '0']
 ['1' '0' '1' '1' '0' '1']
 ['0' '1' '1' '0' '1' '1']
 ['1' '0' '1' '1' '1' '0']
 ['0' '1' '0' '1' '0' '1']
 ['1' '1' '0' '0' '0' '1']
 ['0' '1' '0' '1' '1' '0']
 ['1' '1' '1' '0' '0' '1']
 ['0' '1' '1' '1' '0' '1']
 ['1' '0' '0' '1' '1' '0']
 ['0' '1' '0' '0' '0' '1']
 ['1' '0' '0' '1' '1' '0']
 ['1' '1' '0' '1' '1' '1']
 ['0' '0' '0' '0' '0' '0']
 ['1' '1' '1' '1' '1' '0']
 ['1' '0' '1' '1' '1' '0']
 ['0' '0' '1' '0' '1' '0']
 ['0' '0' '1' '0' '0' '0']
 ['1' '1' '0' '1' '1' '0']
 ['0' '0' '0' '0' '1' '1']
 ['0' '1' '0' '1' '1' '1']
 ['1' '0' '1' '1' '0' '0']
 ['1' '1' '1' '0' '1' '1']
```

```
['1' '0' '0' '1' '1' '0']
['0' '0' '1' '1' '0' '1']
['0' '0' '1' '1' '0' '0']
['1' '1' '1' '0' '1' '1']
['0' '1' '1' '0' '1' '1']
['1' '0' '0' '1' '0' '1']
['0' '0' '1' '1' '1' '0']
['0' '1' '0' '1' '1' '0']
['1' '0' '0' '0' '1' '0']
['0' '0' '0' '0' '1' '0']
['0' '0' '0' '1' '1' '0']
['0' '0' '1' '0' '1' '1']
['0' '1' '0' '1' '0' '1']
['1' '1' '0' '1' '0' '1']
['1' '0' '0' '1' '1' '1']
['0' '0' '0' '0' '0' '0']
['0' '0' '1' '0' '0' '1']
['0' '0' '1' '0' '1' '1']
['0' '1' '1' '1' '0' '0']
['1' '1' '1' '1' '1' '0']
['0' '1' '1' '0' '0' '0']
['0' '1' '0' '0' '0' '1']
['0' '1' '0' '0' '0' '0']
['1' '1' '1' '0' '0' '0']
['1' '1' '0' '1' '0' '0']
['0' '1' '1' '1' '0' '0']
['1' '0' '1' '1' '0' '0']
['0' '1' '1' '0' '1' '1']
['1' '0' '0' '0' '1' '0']
['1' '0' '0' '0' '0' '0']
['0' '0' '0' '0' '0' '1']
['0' '0' '0' '0' '1' '1']
['0' '1' '1' '1' '1' '0']
['1' '1' '1' '1' '0' '0']
['0' '0' '1' '0' '1' '0']
['1' '0' '1' '0' '0' '1']
['1' '0' '0' '0' '1' '0']
['0' '1' '0' '1' '0' '1']
['0' '0' '0' '0' '1' '0']
['0' '0' '0' '0' '0' '1']
['1' '0' '0' '1' '1' '0']
['1' '0' '1' '1' '1' '1']
['1' '0' '0' '0' '0' '1']
['1' '0' '0' '0' '0' '1']
```

```
 ['0' '0' '1' '0' '1' '0']
 ['0' '1' '0' '0' '1' '1']
 ['1' '1' '0' '1' '0' '0']
 ['0' '1' '0' '0' '1' '1']
 ['0' '0' '0' '0' '0' '1']
 ['1' '0' '0' '1' '1' '0']
 ['1' '1' '0' '1' '1' '1']
 ['0' '1' '1' '1' '1' '1']
 ['1' '1' '0' '0' '0' '1']
 ['1' '1' '0' '0' '0' '1']
 ['0' '0' '0' '1' '1' '1']
 ['0' '0' '1' '1' '0' '1']
 ['0' '0' '0' '0' '0' '0']
 ['1' '0' '0' '1' '0' '1']
 ['0' '0' '1' '0' '0' '0']
 ['0' '0' '0' '0' '0' '1']
 ['1' '1' '0' '1' '0' '1']
 ['1' '0' '0' '1' '1' '1']
 ['0' '0' '1' '0' '1' '0']
 ['0' '1' '0' '0' '1' '1']
 ['1' '1' '0' '0' '0' '1']
 ['1' '1' '1' '0' '1' '0']
 ['1' '0' '1' '1' '1' '1']
 ['1' '0' '1' '0' '1' '0']
 ['0' '0' '0' '0' '1' '1']
 ['0' '0' '1' '0' '0' '0']
 ['1' '1' '1' '1' '0' '1']
 ['1' '1' '0' '1' '0' '1']
 ['0' '1' '1' '0' '0' '0']
 ['1' '0' '1' '1' '0' '0']
 ['1' '0' '1' '0' '1' '0']
 ['1' '0' '0' '1' '1' '0']]
0.49
0.51
0.46938775510204084
0.673469387755102
0.46938775510204084
0.4897959183673469
0.40816326530612246
0.5306122448979592
0.32653061224489793
0.5306122448979592
0.5102040816326531
0.5918367346938775
```

```
    0.5294117647058824
    0.45098039215686275
    0.6274509803921569
    0.5686274509803921
    0.5490196078431373
    0.47058823529411764
    0.5490196078431373
    0.37254901960784315
    0.43137254901960786
    0.45098039215686275

Process finished with exit code 0
```

Observation

These code is in line with expectations.