[ML20] Assignment 11

Danny Radosevich

Due: Apr 27 (by 8:45am)

Let us manually build classifiers to predict student performance (Satisfactory/Unsatisfactory) based on three features: major (1=CS/0=non-CS), gender (1=female/0=male) and study hour (0.5/1/2).

Table 1 shows a training set, and Table 2 shows a testing set. Use them to answer the following questions.

You can use calculator/computer to assist the numerical calculations, but you should elaborate the mathematical arguments as much as possible – for example, if you estimate a probability by first decomposing it and then estimating each component, please show your decomposition and the estimates of each component.

If your answer to a question is only a number, you will lose 30% points even if that number is correct.

Table 1. Training Set

Student ID	Major	Gender	Study Hour	Performance
01	1	1	2	S
02	1	1	1	S
03	1	0	2	S
04	0	0	2	S
05	1	1	1	S
06	1	0	0.5	U
07	0	0	0.5	U
08	0	1	1	U
09	1	0	1	U
10	1	1	0.5	U

Table 2. Testing Set

Student ID	Major	Gender	Study Hour	Performance
11	0	1	2	?
12	0	1	0.5	?

[1] Manually learn a Naive Bayes classifier from the training set and apply it to classify the testing set. Please complete the following steps.

(1.a) Elaborate your estimations of $p(y = S \mid x)$ and $p(y = U \mid x)$, and highlight your results. $p(y = S \mid x)$ and $p(y = U \mid x)$ where $x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$

$p(y = S \mid x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix})$ and $p(y = U \mid x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix})$

$(x_0 \perp x_1 \perp x_2 \mid y)$

$p(x = x_1, x = x_2, x = x_3 \mid y = S) = p(y = S) \cdot p(x = x_1 \mid y = S) \cdot p(x = x_2 \mid y = S) \cdot p(x = x_3 \mid y = S)$

$p(x = x_1, x = x_2, x = x_3 \mid y = U) = p(y = U) \cdot p(x = x_1 \mid y = U) \cdot p(x = x_2 \mid y = U) \cdot p(x = x_3 \mid y = U)$

(1.b) Clearly state your predicted classes for students 11 and 12.

0.1 student 11

$p(x = 0, x = 1, x = 2 \mid y = S) = p(y = S) \cdot p(x = 0 \mid y = S) \cdot p(x = 1 \mid y = S) \cdot p(x = 2 \mid y = S)$

$p(x = 0, x = 1, x = 2 \mid y = S) = (1/2 \cdot 1/5 \cdot 3/5 \cdot 3/5) = 9/250$

$p(x = 0, x = 1, x = 2 \mid y = U) = p(y = U) \cdot p(x = 0 \mid y = U) \cdot p(x = 1 \mid y = U) \cdot p(x = 2 \mid y = U)$

$p(x = 0, x = 1, x = 2 \mid y = S) = (1/2 \cdot 2/5 \cdot 2/5 \cdot 0/5) = 0$

student 11 would be successful.

0.2 student 12

$p(x = 0, x = 1, x = .5 \mid y = S) = p(y = S) \cdot p(x = 0 \mid y = S) \cdot p(x = 1 \mid y = S) \cdot p(x = .5 \mid y = S)$

$p(x = 0, x = 1, x = .5 \mid y = S) = (1/2 \cdot 1/5 \cdot 3/5 \cdot 0/5) = 0$

$p(x = 0, x = 1, x = .5 \mid y = U) = p(y = U) \cdot p(x = 0 \mid y = U) \cdot p(x = 1 \mid y = U) \cdot p(x = .5 \mid y = U)$

$p(x = 0, x = 1, x = .5 \mid y = S) = (1/2 \cdot 2/5 \cdot 2/5 \cdot 3/5) = 12/250$

student 12 would be unsuccessful.

[2] Manually learn a Linear Discriminant Analysis classifier from the training set and apply it to classify the testing set. Please complete the following steps.

(2.a) Elaborate your estimations of $p(y = S \mid x)$ and $p(y = U \mid x)$, and highlight your results.

0.3 successful:

$$\text{muS} = \begin{bmatrix}(1+1+1+0+1)/5\\(1+1+0+0+1)/5\\(2+1+2+2+1)/5\end{bmatrix} = \begin{bmatrix}4/5\\3/5\\8/5\end{bmatrix}$$

0.4 unsuccessful

$$\text{muU} = \begin{bmatrix}(1+0+0+1+1)/5\\(0+0+1+0+1)/5\\(.5+.5+1+1+.5)/5\end{bmatrix} = \begin{bmatrix}3/5\\2/5\\7/10\end{bmatrix}$$

(2.b) Clearly state your predicted classes for students 11 and 12.

0.5 math

Using the discriminate equation $f = mu \cdot C^{-1} \cdot x^T - (1/2)\mu \cdot C^{-1} \cdot \mu^T + ln(p)$ We can use this equation 4 times.
p will always equal 1/2
Twice with x = features of student 11 with the two μ's found above
Twice with x = features of student 12 with the two μ's found above

0.6 code

I used the following code to find the values:

```
sele = np.asarray(sele)
stwe = np.asarray(stwe)
muS = np.array([4/5,3/5,8/5])
muU = np.array([3/5,2/5,7/10])
cov = np.identity(3)
cov = np.linalg.inv(cov)
daMus =[muS,muU]
for mu in daMus:
    partOne = np.dot(mu,cov)
    partOne = np.dot(partOne,np.transpose(sele))
    partTwo = np.dot(mu,cov)
    partTwo = np.dot(partTwo, np.transpose(mu))
    partTwo = np.dot((1/2),partTwo)
    print(partOne-partTwo+np.log(1/2))
for mu in daMus:
    partOne = np.dot(mu,cov)
    partOne = np.dot(partOne,np.transpose(stwe))
    partTwo = np.dot(mu,cov)
    partTwo = np.dot(partTwo, np.transpose(mu))
    partTwo = np.dot((1/2),partTwo)
    print(partOne-partTwo+np.log(1/2))
```

0.7 classification

The values found were:
successful mu for student 11 : 1.3268528194400546
unsuccessful mu for student 11 : 0.6018528194400546
So student 11 should be classified as successful
successful mu for student 12 : -1.0731471805599457
unsuccessful mu for student 12 : -0.4481471805599453
So student 11 should be classified as unsuccessful

[3] Manually learn a k-NN classifier from the training set and apply it to classify the testing set. Please complete the following steps.

(3.a) Show the distances between training and testing instances in Table 3. For example, X is the distance between students 11 and 04, and Y is the distance between 12 and 02. Use L2-distance (not squared).

0.8 WORK: The following code was used to find L2-Distance

```
#Danny Radosevich
import numpy as np

students = {}
students[1] = [1,1,2]
students[2] = [1,1,1]
students[3] = [1,0,2]
students[4] = [0,0,2]
students[5] = [1,1,1]
students[6] = [1,0,.5]
students[7] = [0,0,.5]
students[8] = [0,1,1]
students[9] = [1,0,1]
students[10] = [1,1,.5]

sele = [0,1,2]
stwe = [0,1,.5]

for stu in students:
    print("sele",stu)
    a = np.asarray(sele)
    b = np.asarray(students[stu])
    print(np.linalg.norm(a-b))
print("-------------------------------------")
for stu in students:
    print("stwe",stu)
    a = np.asarray(stwe)
    b = np.asarray(students[stu])
    print(np.linalg.norm(a-b))
```

Table 3. Pair-wise Distance Measure

Student	01	02	03	04	05	06	07	08	09	10
11	1.0	1.41	1.41	1.0	1.41	2.06	1.8	1.0	1.73	1.8
12	1.8	1.12	2.06	1.8	1.12	1.41	1.0	0.5	1.5	1.0

(3.b) Identify the K nearest neighbors and their labels of student 11 in Table 5. For example, suppose student 03 is the 4th nearest neighbor of student 11, then X = 03 and Y = S.

Table 4. K-nearest neighbor of Student 11

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Neighbor ID	1	4	8	2	3	5	9	7	10	6
Label	S	S	U	S	S	S	U	U	U	U

Table 5. K-nearest neighbor of Student 12

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Neighbor ID	8	7	10	2	5	6	9	1	8	3
Label	U	U	U	S	S	U	U	S	U	S

(3.c) Show your classification results on testing students in Table 6. For example, X is result of student 11 when K = 3, and Y is result of 12 when K = 5.

Table 6. Classification Results

Student IC	K=1	K=2	K=3	K=4	K=5
11	S	S	S	S	S
12	U	U	U	U	U

[4] (Bonus) Implement k-NN with L2-distance and evaluate it on the COMPAS data set. Please draw a curve of testing error versus K in Figure 1 – its x-axis is K and y-axis is classification error on the testing set.[1] Choose 5 values of K yourself so that your curve can cover a relatively complete story.

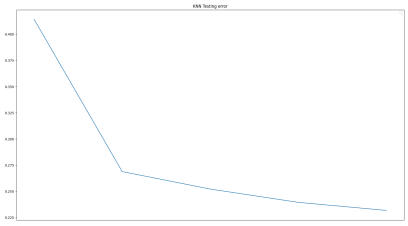

Fig. 1. Testing Error of kNN versus K.

[1] Keep in mind that testing error is the fraction of mis-classified testing instances.

[5] (Bonus) Redo task [4], but this time implement a kernel kNN (kkNN) instead. Choose the kernel function and its hyper-parameters yourself. Similar to Figure 1, draw two error curves in Figure 2 – one curve of kNN and one curve of kkNN. (If kkNN outperforms kNN at one value of K, you get a 10% additional bonus for this problem. If kkNN outperforms kNN at all five values of K...)

Fig. 2. Testing Errors of kNN and kkNN versus K.