



INTRODUCTION TO MACHINE LEARNING

First Homework

Julian Ewaied and Namir Ballan



CONTENTS

Question 2.....	2
Question 4.....	2
Question 5.....	4

QUESTION 2

Let S be the event of the tested person being sick, and T be the event of the test being positive. We're given a prior $P(S) = \frac{1}{1000}$. We're also given the conditional probability $P(T|\bar{S}) = 0.01$, $P(\bar{T}|S) = 0$.

- a. We want to calculate the conditional probability $P(S|T)$. We can use Bayes rule for that:

$$P(S|T) = P(T|S) \cdot \frac{P(S)}{P(T)} = (1 - P(\bar{T}|S)) \cdot \frac{P(S)}{P(T|\bar{S})P(\bar{S}) + P(T|S)P(S)}$$
$$P(S|T) = 1 \cdot \frac{10^{-3}}{0.01 \cdot (1 - 0.001) + 1 \cdot 0.001} = 0.091$$

- b. It is more probable I don't have the disease since $P(S|T) < P(\bar{S}|T)$.
- c. Obviously yes, since the maximum likelihood would ignore the strong prior we have, which makes it almost sure that we have the disease (the likelihood $P(T|S) = 1 - P(\bar{T}|S) = 1$).

QUESTION 4

We wanted to implement a naïve Bayes classifier using MAP. For that purpose, we had a huge dataset of labels and sentences. We defined a class 'NBClassifier'. The constructor of the class receives an array-like object where each entry of it is an array of words (split string), called `texAll`, and respectively the sentences' label, called `lbAll`. In addition, we had all possible categories and our vocabulary. Then, we built counts – the number of appearances of each label, in a dictionary, and totals – a dictionary to save the number of words for each label.

Then, we apply `train()`, which learns the data by finding the class conditionals $P(x|\omega_i)$, and the priors $P(\omega_i)$ using MLE as we proved in class. we apply logarithms on all numbers to avoid floating point underflow, and for convenience we do it with base 2 (because the smaller the base, the more precise the numbers).

Now that we have the class conditionals and priors, we can write '`calc_posterior()`', which receives a sentence and a label, and calculates the log of the posterior

$$\log(P(label)P(sentence|label)) = \log(P(label)) + \sum_{w \in sentence} \log(P(w|label))$$

Keeping in consideration that some words might not be in the vocabulary or might not appear in some label, and therefore we should use 'totals' to calculate the answer with Laplace smoothing (only for words whose probability is 0). The probability of a word with a zero-probability is:

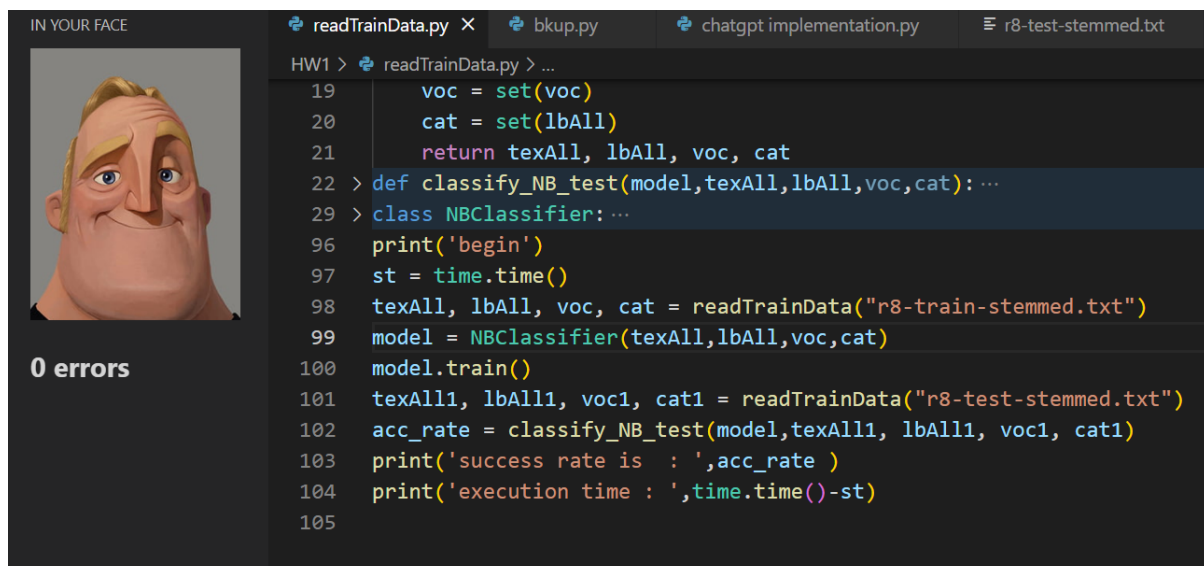
$$P(\text{word}|\text{label}) = \frac{1}{N_{\text{label}} + N_{\text{voc}}}$$

This makes sense since the more we know words, the less it's probable to get a new word, and the more we've seen words in a label, the less it's probable to miss a word. Therefore, we've minimized the effect of new words but still didn't ignore them.

And then, classify can make use of them to calculate the MAP output. The final step is to test the classifier, which can be done using the test dataset, which is done by the function 'NB_classify_test()', as we read from the test file and compare our classifier's answer to the actual answer. Eventually we receive the success rate and print it, which is in this case and for the input we've received:

```
success rate is : 0.9643672910004568
execution time : 1.3759894371032715
```

Which is:



The screenshot shows a Jupyter Notebook with a dark theme. On the left, there is a panel titled 'IN YOUR FACE' containing a cartoon image of Mr. Incredible and the text '0 errors'. The main area displays a code editor with the following Python code:

```
HW1 > readTrainData.py > ...
19     voc = set(voc)
20     cat = set(lbAll)
21     return texAll, lbAll, voc, cat
22 > def classify_NB_test(model, texAll, lbAll, voc, cat): ...
29 > class NBClassifier: ...
96     print('begin')
97     st = time.time()
98     texAll, lbAll, voc, cat = readTrainData("r8-train-stemmed.txt")
99     model = NBClassifier(texAll, lbAll, voc, cat)
100    model.train()
101    texAll1, lbAll1, voc1, cat1 = readTrainData("r8-test-stemmed.txt")
102    acc_rate = classify_NB_test(model, texAll1, lbAll1, voc1, cat1)
103    print('success rate is : ', acc_rate)
104    print('execution time : ', time.time()-st)
105
```

Figure 1: very good!!

For more explanation check the attached code.

QUESTION 5

Let's look at the output of the Bayesian classifier:

$$c_i = \arg \max_{\alpha_i} -R(\alpha_i|x) = \arg \max_{\alpha_i} - \sum_{c_j} \lambda(\alpha_i|c_j)P(c_j|x) = \arg \max_{\alpha_i} - \sum_{i \neq j} P(c_j|x)$$

Where the last transition is simply substituting the loss function in.

$$c_i = \arg \max_{\alpha_i} \left(1 - \sum_{i \neq j} P(c_j|x) \right) = \arg \max_{\alpha_i} (P(c_i|x))$$

We only added 1 which doesn't change the argument maximum, and reached a new probability argmax which is the MAP output.

Q1

let S be the event where we got a share picture, and D be the event where we got a diminished picture.

let G be the event where the TV is good, F the event where its fair, and B the event where it bad.

first lets calculate $P(S)$ and $P(D)$:

$$\begin{aligned} P(S) &= P(S|G)P(G) + P(S|F)P(F) + P(S|B)P(B) \\ &= 0.9 * 0.3 + 0.5 * 0.5 + 0.2 * 0.2 = 0.56 \end{aligned}$$

$$P(D) = 1 - P(S) = 1 - 0.56 = 0.44$$

now lets calculate all of the pairs of $P(TV\ type|image\ type)$ using bayes rule:

$P(G|S)$:

$$P(G|S) = \frac{P(S|G)P(G)}{P(S)} = \frac{0.9*0.3}{0.56} = 0.482$$

$P(G|D):$

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} = \frac{0.1*0.3}{0.44} = 0.068$$

$P(F|S):$

$$P(F|S) = \frac{P(S|F)P(F)}{P(S)} = \frac{0.5*0.5}{0.56} = 0.446$$

$P(F|D):$

$$P(F|D) = \frac{P(D|F)P(F)}{P(D)} = \frac{0.5*0.5}{0.44} = 0.568$$

$P(B|S):$

$$P(B|S) = \frac{P(S|B)P(B)}{P(S)} = \frac{0.2*0.2}{0.56} = 0.07$$

$P(B|D):$

$$P(B|D) = \frac{P(D|B)P(B)}{P(D)} = \frac{0.8*0.2}{0.44} = 0.363$$

lets denote with α_1 the decision where we buy the TV
and

α_2 the decision where we dont buy it.

now lets calculate all of the pairs of $R(\text{decision}|\text{image type})$:

$$R(\alpha_1|S)$$

$$\begin{aligned} R(\alpha_1|S) &= P(G|S)*\lambda(\alpha_1|G) + P(F|S)*\lambda(\alpha_1|F) + P(B|S)*\lambda(\alpha_1|B) \\ &= 0.482*0 + 0.446*5 + 0.07*20 = 3.63 \end{aligned}$$

$$R(\alpha_1|D)$$

$$\begin{aligned} R(\alpha_1|D) &= P(G|D)*\lambda(\alpha_1|G) + P(F|D)*\lambda(\alpha_1|F) + P(B|D)*\lambda(\alpha_1|B) \\ &= 0.068*0 + 0.568*5 + 0.363*20 = 10.1 \end{aligned}$$

$$R(\alpha_2|S)$$

$$\begin{aligned} R(\alpha_2|S) &= P(G|S)*\lambda(\alpha_2|G) + P(F|S)*\lambda(\alpha_2|F) + P(B|S)*\lambda(\alpha_2|B) \\ &= 0.482*10 + 0.446*5 + 0.07*0 = 7.05 \end{aligned}$$

$$R(\alpha_2|D)$$

$$\begin{aligned} R(\alpha_2|D) &= P(G|D)*\lambda(\alpha_2|G) + P(F|D)*\lambda(\alpha_2|F) + P(B|D)*\lambda(\alpha_2|B) \\ &= 0.068*10 + 0.568*5 + 0.363*0 = 3.52 \end{aligned}$$

as a result of $R(\alpha_1|S) < R(\alpha_2|S)$ we get that the optimal decision to make if the image is sharp is to buy the TV.

as a result of $R(\alpha_2|D) > R(\alpha_1|D)$ we get that the optimal decision to make if the image is not sharp is to not

buy the TV.

lets calculate the total risk:

$$\begin{aligned} R(\alpha) &= R(\alpha_1|S)P(S) + R(\alpha_2|D)P(D) = \\ &3.63*0.56 + 3.52*0.44 = 3.581 \end{aligned}$$

Q3

A:

lets denote with a_j the frequency of number j in dice 1,
and with b_j the frequency of the number j in dice 2.

lets now calculate $P_i(j)$. lets assume w.l.o.g that $i=1$.
we are trying to estimate the probability. so lets
calculate the likelihood for some value p :

given that the probability is p , the distribution of
 a_j is
 $\text{Bin}(40, p)$. and so we get

$L(p) = \binom{40}{a_j} p^{a_j} (1-p)^{40-a_j}$. to simplify, lets calculate using
log-likelihood:

$$\log(L(p)) = \log\left(\binom{40}{a_j} p^{a_j} (1-p)^{40-a_j}\right) = \log\left(\binom{40}{a_j}\right) + a_j \log p + (40 - a_j) \log(1-p)$$

lets now derive $\log(L(p))$ and find a p that makes the derivative 0:

$$\log(L(p))' = \frac{a_j}{p} - \frac{40 - a_j}{1-p} = 0$$

$$\implies a_j(1-p) - (40 - a_j)p = 0$$

$$\implies a_j - pa_j - 40p + a_jp = 0$$

$$\implies a_j = 40p \implies p = \frac{a_j}{40}$$

so we found that the MLE estimate for $P_i(j)$ is $\frac{\text{frequency of } j \text{ in dice } i}{40}$.

so now according to this rule lets calculate $P_i(j)$ for all of the values of i, j :

i \ j	1	2	3	4	5	6
1	$\frac{1}{8}$	$\frac{3}{40}$	$\frac{1}{4}$	$\frac{1}{40}$	$\frac{1}{4}$	$\frac{11}{40}$
2	$\frac{1}{4}$	$\frac{11}{40}$	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{3}{40}$	$\frac{1}{20}$

B:

lets denote with c_1 the event where the dice is the first dice and c_2 the event where the dice is the second dice.

let x denote the known 40 dice rolls.

$$P(c_1|x) = \frac{P(x|c_1)p(c_1)}{P(x)}, P(c_2|x) = \frac{P(x|c_2)p(c_2)}{P(x)}$$

we want to find the i that maximises $P(c_i|x)$. because in both cases we have devision by $P(x)$, then the i that maximises $P(c_i|x)$ is also the i that maximises $P(x|c_i)P(c_i)$. $P(c_i) = \frac{1}{2}$ because we have no prior about which dice is more likely to be lost, and so we want the i that maximises $P(x|c_i)$, which is the MLE estimate.

lets calculate $P(x|c_i)$ for each i :

$P(x|c_1)$:

to calculate the probability of getting those exact dice rolls, we will calculate the number of possible permutations for x , and multiply that for the probability of getting that permutation. its clear to see that all the permutations have an equal probability of happening so all we need to do it calculate the probability of a permutation.

lets look at the permutation where we first get all

of the 1's, and then 2's and then 3's and so on.

the probability of this permutation is

$$\begin{aligned}
 & P_1(1)^8 * P_1(2)^{12} * P_1(2)^6 * P_1(4)^9 * P_1(5)^4 * P_1(6) = \\
 & = \left(\frac{1}{8}\right)^8 * \left(\frac{3}{40}\right)^{12} * \left(\frac{1}{4}\right)^6 * \left(\frac{1}{40}\right)^9 * \left(\frac{1}{4}\right)^4 * \left(\frac{11}{40}\right) = 1.88 * 10^{-42}
 \end{aligned}$$

lets denote the number of permutations with p .

and so we get that $P(x|c_1) = p * 1.88 * 10^{-42}$.

$P(x|c_2)$:

we will calculate the probability of each permutation in the same way:

$$\begin{aligned}
 & P_2(1)^8 * P_2(2)^{12} * P_2(2)^6 * P_2(4)^9 * P_2(5)^4 * P_2(6) = \\
 & = \left(\frac{1}{4}\right)^8 * \left(\frac{11}{40}\right)^{12} * \left(\frac{1}{10}\right)^6 * \left(\frac{1}{4}\right)^9 * \left(\frac{3}{40}\right)^4 * \left(\frac{1}{20}\right) = 1.722 * 10^{-29}
 \end{aligned}$$

lets denote the number of permutations with p .

and so we get that $P(x|c_2) = p^* 1.722 \cdot 10^{-29}$.

$P(x|c_2) = p^* 1.722 \cdot 10^{-29} > p^* 1.88 \cdot 10^{-42} = P(x|c_1)$ and so the MLE estimate is c_2 . we explained earlier why the MLE estimate is the optimal bayes estimate in this case.