Introduction to Data Mining: 21191441
Zhejiang University — April 27, 2020
Professor Deng Cai — Homework 1

Homework 1

Collaborators:
Name: Jiang Shibiao
Student ID: 3170102587

Problem 1-1. Machine Learning Problems

(a) Choose proper word(s) from

Answer:

1. BF
2. C
3. AD
4. BG
5. AE
6. AD
7. BF
8. AE
9. BG

(b) True or False: To fully utilizing available data resource, we should use all the data we have to train our learning model and choose the parameters that maximize performance on the whole dataset. Justify your answer.

Answer: False. Generally speaking, the more data we use and we try to fit, then the more complex the model is, and complex model needs larger amounts of parameters which will easily cause overfitting. Usually we divide the labeled data into two parts: *train part* and *validation part*.

Problem 1-2. Bayes Decision Rule

(a) Suppose you are given a chance to win bonus grade points:

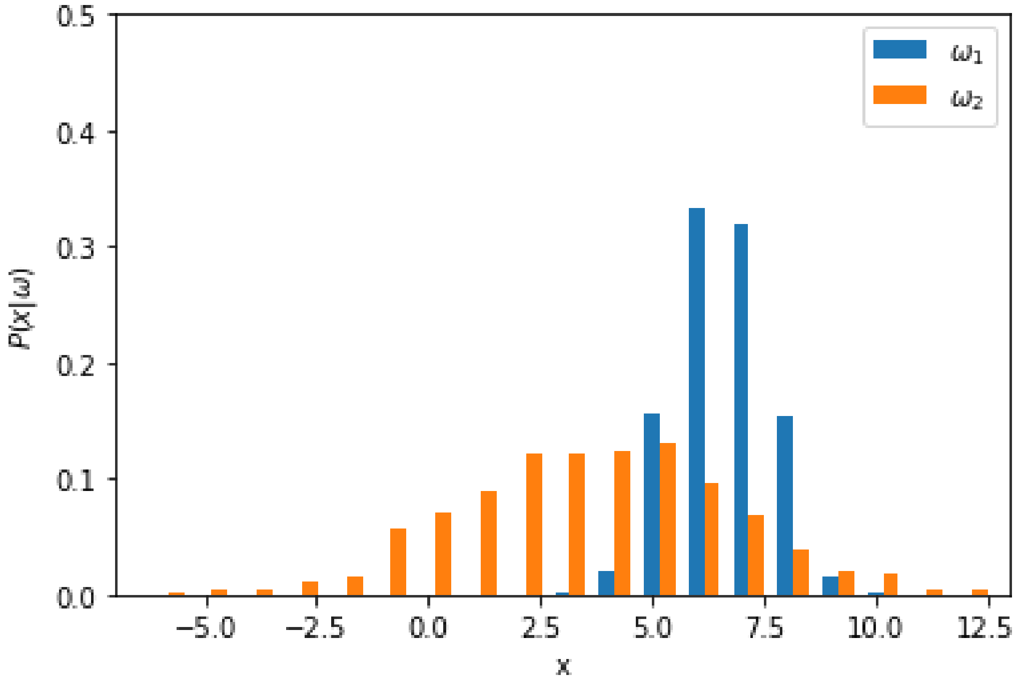
Answer:

1. $P(B_1 = 1) = \frac{1}{3}$
2. $P(B_2 = 0|B_1 = 1) = 1$
3. $P(B_1 = 1|B_2 = 0) = \frac{P(B_2=0|B_1=1)P(B_1=1)}{P(B_2=0)} = \frac{1}{3}$
4. Change my choice.

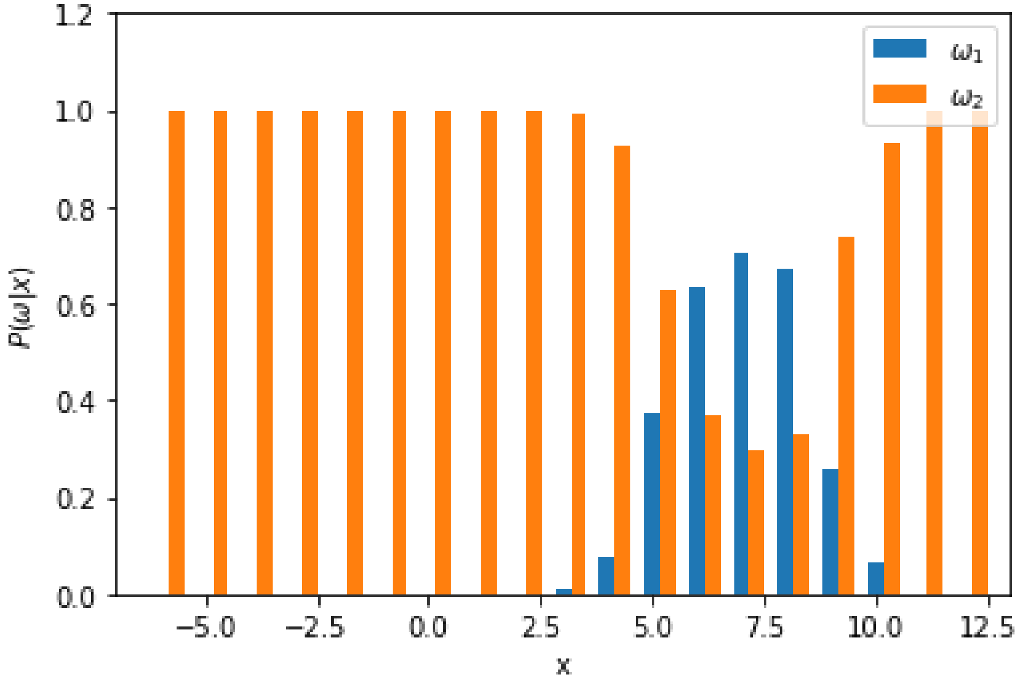
(b) Now let us use bayes decision theorem to make a two-class classifier $\cdots$.

Answer:

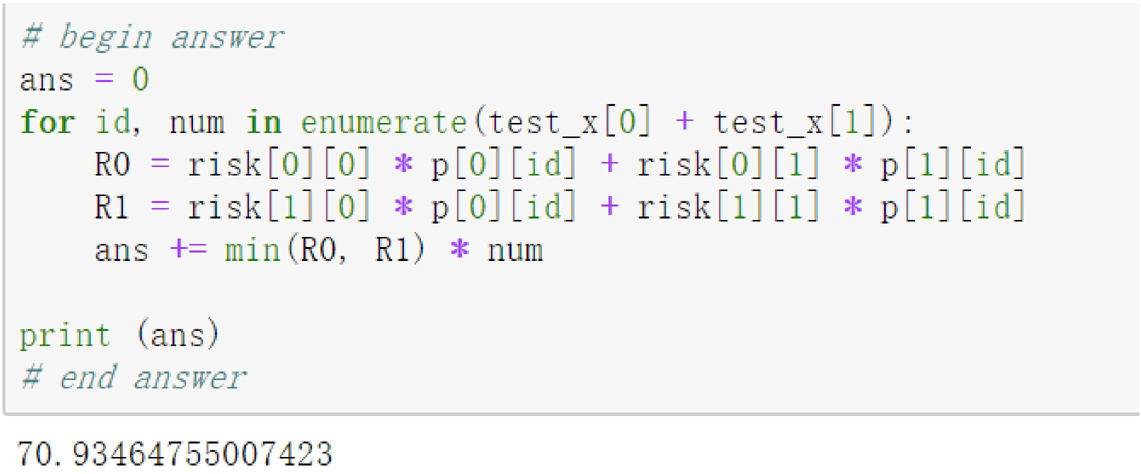
1. Error rate: 0.21333333333333333



2. Error rate: 0.15666666666666667



3. The loss is 70.93.



```
# begin answer
ans = 0
for id, num in enumerate(test_x[0] + test_x[1]):
    R0 = risk[0][0] * p[0][id] + risk[0][1] * p[1][id]
    R1 = risk[1][0] * p[0][id] + risk[1][1] * p[1][id]
    ans += min(R0, R1) * num

print (ans)
# end answer
```

```
70.93464755007423
```

Problem 1-3. Gaussian Discriminant Analysis and MLE

Given a dataset consisting of m samples. We assume these samples are independently generated by one of two Gaussian distributions$\cdots$

(a) What is the decision boundary?

Answer:

$$
\begin{aligned}
p(y=1|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=1)\times p(y=1)}{p(\boldsymbol{x})} \\
p(\boldsymbol{x}|y=1) &= \frac{1}{2\pi}e^{-\frac{1}{2}((x_1-1)^2+(x_2-1)^2)} = 2 \\
p(y=1) &= \phi \\
p(\boldsymbol{x}) &= p(\boldsymbol{x}|y=1)p(y=1) + p(\boldsymbol{x}|y=0)p(y=0) \\
&= \frac{1}{2\pi}e^{-\frac{1}{2}((x_1-1)^2+(x_2-1))^2)}\phi + \frac{1}{2\pi}e^{-\frac{1}{2}(x_1^2+x_2^2)}(1-\phi) \\
p(y=1|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=1)\times p(y=1)}{p(\boldsymbol{x})} \\
&= \frac{\frac{1}{2\pi}e^{-\frac{1}{2}((x_1-1)^2+(x_2-1))^2)}\phi}{\frac{1}{2\pi}e^{-\frac{1}{2}((x_1-1)^2+(x_2-1))^2)}\phi + \frac{1}{2\pi}e^{-\frac{1}{2}(x_1^2+x_2^2)}(1-\phi)} \\
&= \frac{\phi e^{1-(x_1+x_2)}}{\phi e^{1-(x_1+x_2)} + (1-\phi)} \\
p(y=0|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=0)\times p(y=0)}{p(\boldsymbol{x})} \\
&= \frac{1-\phi}{\phi e^{1-(x_1+x_2)} + (1-\phi)}
\end{aligned}
$$

$p(y=1|\boldsymbol{x}) = p(y=0|\boldsymbol{x})$, so the decision boundary is $x_1+x_2 = 1-\ln(1-\phi)+\ln(\phi))$. We know $\phi = \frac{1}{2}$, then the decision boundary is $x_1 + x_2 = 1$

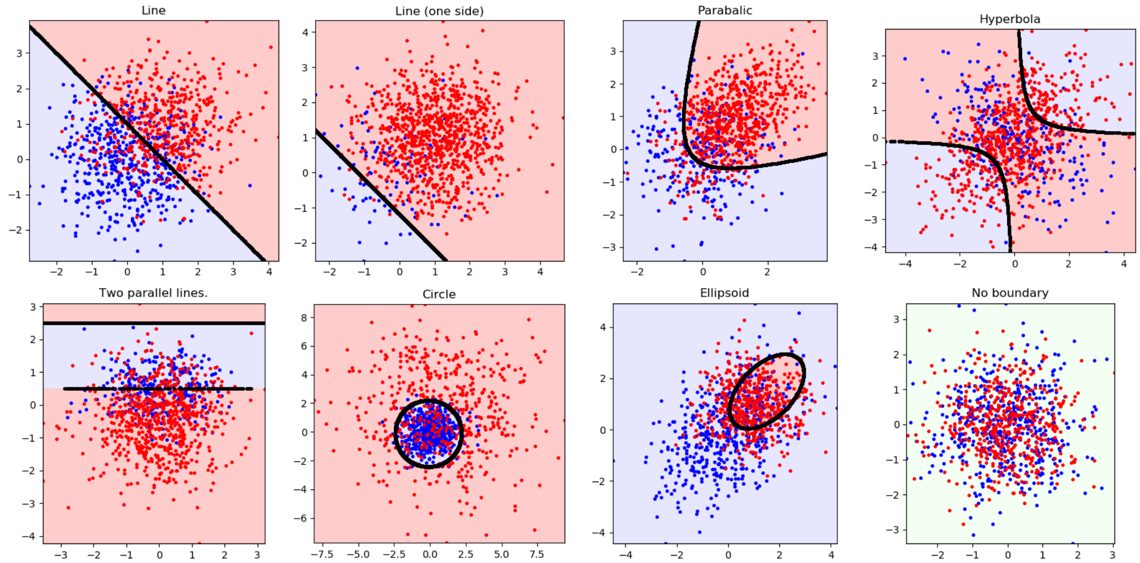
(b) An extension of the above model is to classify K classes by fitting a Gaussian distribution for each class$\cdots$

Answer:

Please find it in gaussian_ pos_ prob.py.

(c) Now let us do some field work playing with the above 2-class Gaussian discriminant model.

Answer:



(d) What is the maximum likelihood estimation of ϕ, μ_0 and μ_1?

Answer:

$$L = \prod_{y_i=0} (1-\phi) N(\mu_0, \Sigma_0, \boldsymbol{x}_i) \cdot \prod_{y_i=1} \phi N(\mu_1, \Sigma_1, \boldsymbol{x}_i).$$

Denote $n_i (i = \{0, 1\})$ as the number of points for y_i. i.e. $n_i = |\{k | y_k = i\}|$.

$$l = \ln L = n_0 \ln(1-\phi) + n_1 \ln \phi + \sum_{y_i=0} \ln(N(\mu_0, \Sigma_0, \boldsymbol{x}_i)) + \sum_{y_i=1} \ln(N(\mu_1, \Sigma_1, \boldsymbol{x}_i)).$$

For ϕ, let $\frac{\partial l}{\partial \phi} = 0$. So $\frac{n_0}{1-\phi} = \frac{n_1}{\phi}$, finally $\widehat{\phi} = \frac{n_1}{n_0+n_1}$.

For μ_0, let $\frac{\partial l}{\partial \mu_0} = 0$, then $\Sigma_0^{-1} \cdot \sum_{y_i=0} (\boldsymbol{x}_i - \mu_0) = 0$, finally $\widehat{\mu}_0 = \frac{1}{n_0} \sum_{y_i=0} \boldsymbol{x}_i$.

Similarly, $\widehat{\mu}_1 = \frac{1}{n_1} \sum_{y_i=1} \boldsymbol{x}_i$.

Generalize to the K-class gaussian model:

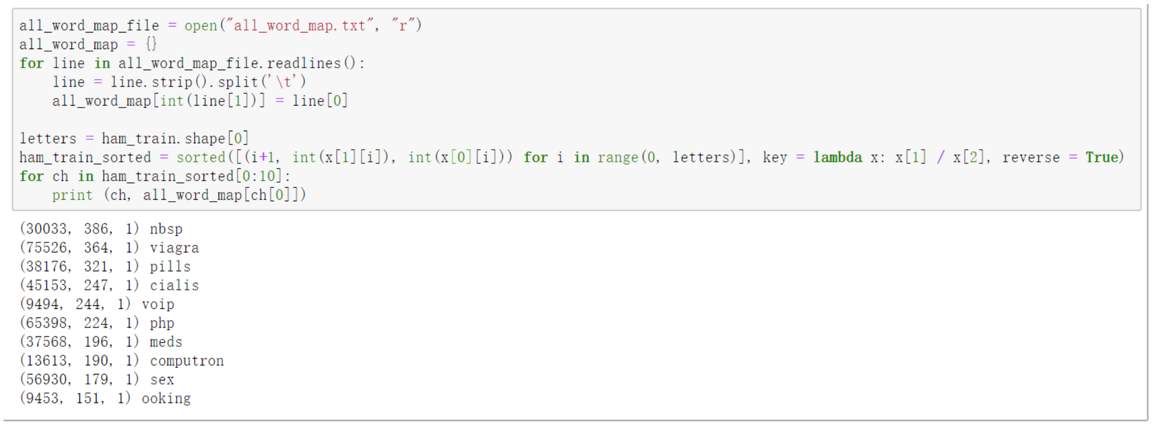
$$\widehat{\phi}_k = \frac{n_k}{\sum n_i}$$

$$\widehat{\mu}_k = \frac{1}{n_k} \sum_{y_i=k} \boldsymbol{x}_k.$$

Problem 1-4. Text Classification with Naive Bayes

(a) List the top 10 words.

Answer:



```
all_word_map_file = open("all_word_map.txt", "r")
all_word_map = {}
for line in all_word_map_file.readlines():
    line = line.strip().split('\t')
    all_word_map[int(line[1])] = line[0]

letters = ham_train.shape[0]
ham_train_sorted = sorted([(i+1, int(x[1][i]), int(x[0][i])) for i in range(0, letters)], key = lambda x: x[1] / x[2], reverse = True)
for ch in ham_train_sorted[0:10]:
    print (ch, all_word_map[ch[0]])
```

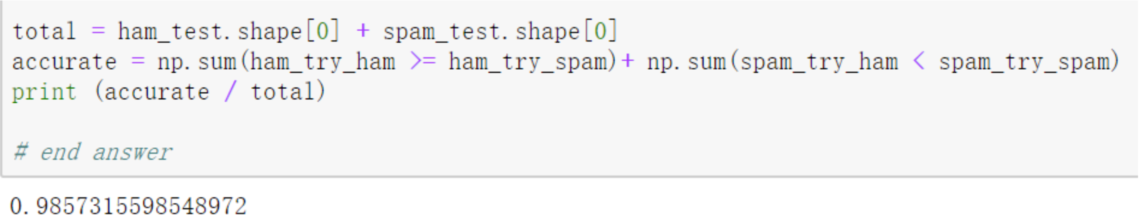
```
(30033, 386, 1) nbsp
(75526, 364, 1) viagra
(38176, 321, 1) pills
(45153, 247, 1) cialis
(9494, 244, 1) voip
(65398, 224, 1) php
(37568, 196, 1) meds
(13613, 190, 1) computron
(56930, 179, 1) sex
(9453, 151, 1) ooking
```

From left to right: ID of word, times in SPAM, times in HAM, word.

Sorted from top to bottom.

(b) What is the accuracy of your spam filter on the testing set?

Answer:



```
total = ham_test.shape[0] + spam_test.shape[0]
accurate = np.sum(ham_try_ham >= ham_try_spam)+ np.sum(spam_try_ham < spam_try_spam)
print (accurate / total)

# end answer
```

```
0.9857315598548972
```

98.6%

(c) True or False: a model with 99% accuracy is always a good model. Why?

Answer:

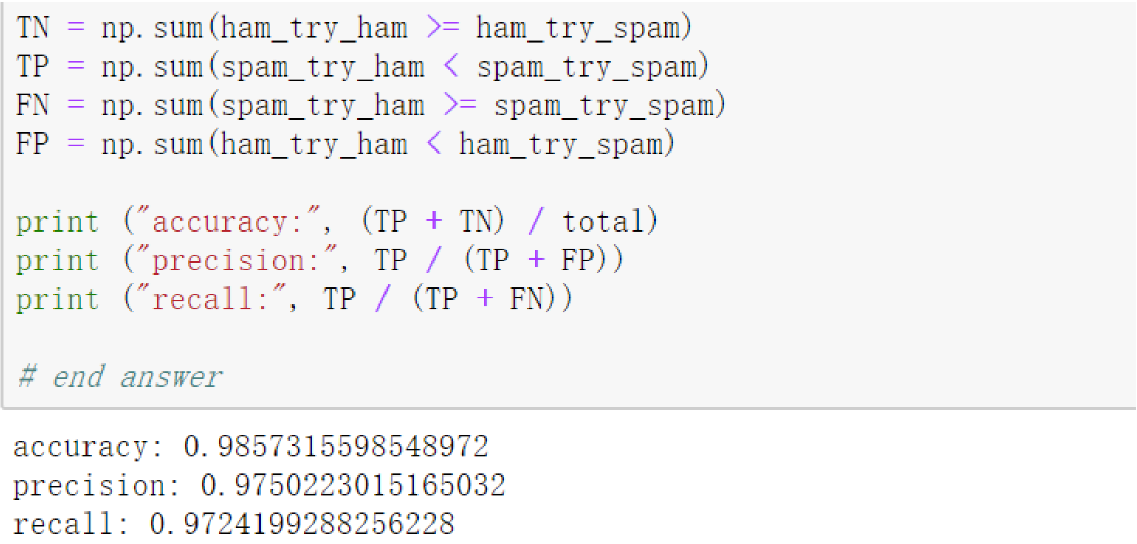
False.

If $P(spam) \sim P(ham)$, this statement is kind of reasonable.

But if $P(spam) << P(ham)$(like $1:99$), this statement is really ridiculous. Consider a naive model: Take all email as the ham. This model reaches "99% accuracy" but is still a terrible one.

(d) Compute the precision and recall of your learnt model.

Answer:



```
TN = np.sum(ham_try_ham >= ham_try_spam)
TP = np.sum(spam_try_ham < spam_try_spam)
FN = np.sum(spam_try_ham >= spam_try_spam)
FP = np.sum(ham_try_ham < ham_try_spam)

print ("accuracy:", (TP + TN) / total)
print ("precision:", TP / (TP + FP))
print ("recall:", TP / (TP + FN))

# end answer
```

```
accuracy: 0.9857315598548972
precision: 0.9750223015165032
recall: 0.9724199288256228
```

(e) For a spam filter, which one do you think is more important, precision or recall? What about a classifier to identify drugs and bombs at airport? Justify your answer.

Answer:

For a spam filter, **precision** is more important. because if we recognize a normal email as spam (i.e. FP), the user may miss this email and will lead a lot of risks.

For a classifier to identify drugs and bombs, **recall** is more important. If we mistakenly identify a innocent people, it's okay and just consume some times. But if we don't find out a crime(i.e. FN) and miss him, he will bring dangers to the airport.