Comp560 Jiabing Song/Yingying Wu

HW3 Report

Bayes Nets writteng questions

A. 1. No because B and I and M are not independent

2. Yes because I and J are independent given G

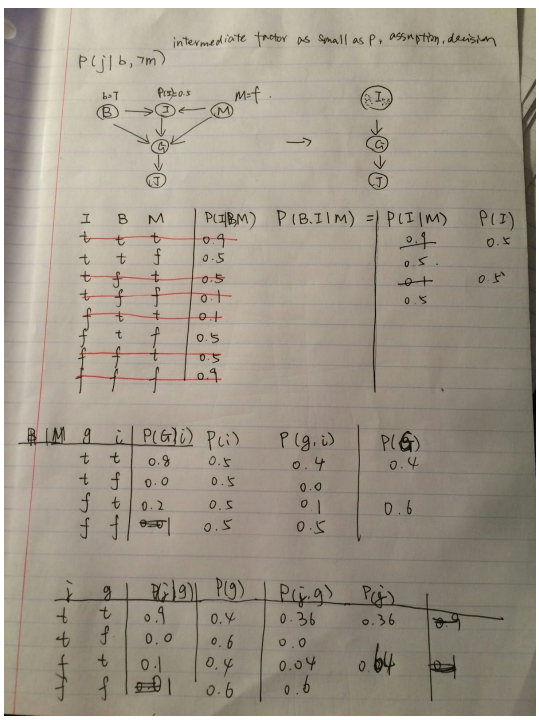
$$P(J|G,I) = \frac{P(J,G,I)}{P(G,I)} = \frac{P(J,I|G)\cdot P(G)}{P(G,I)} = \frac{P(J|G)\cdot P(I|G)\cdot P(G)}{P(G,I)} = \frac{P(J|G)\cdot P(G,I)}{P(G,I)} = P(J|G)$$

3. Yes because M and J are indepenent given G

$$P(M|G,B,I,J) = \frac{P(M,G,B,I,J)}{P(G,B,I,J)} = \frac{P(M,J|G,B,I)\cdot P(G,B,I)}{P(G,B,I,J)} = \frac{P(M|G,B,I)\cdot P(J|G,B,I)\cdot P(G,B,I)}{P(G,B,I,J)} = \frac{P(M|G,B,I)\cdot P(G,B,I,J)}{P(G,B,I,J)} = P(M|G,B,I)$$

B. P(b,i,¬m,g,j) = P(B)*(1-P(M))*P(I|B, ¬M)*P(G|I,B,¬M)*P(J|G)
= 0.9 * 0.9 * 0.5 * 0.8 * 0.9 = 0.2916

C. First we try to join B and M into I because we know that B is true and M is false. From the chart we can sum that P(+I) = 0.5, P(-I) = 0.5 given +B and -M. Next we join into G and we get P(+G) = 0.4, P(-G) = 0.6 after elimination. Finally we join G into J and we get the answer that P(+J) = 0.36 and P(-J) = 0.64. Please see a more detailed describtion from the photo we attached below.



D. We decide to go with likelyhood weighting. Because G is already given, forward sampling is not an option, in that case we will generate a lot of samples that don't meet the requirement. We pick likelyhood weighting over rejecting sampling because we might generate a lot of examples that are inconsistent to the true joint.

Our process will be to generate a list of random numbers and check which category does the number fall into (+ or -) based on the probability and sum the probabilities that match query value together, then divided by the total weights.

For example, we generate some random numbers [0.7,0.62,0.24,0.44]. We fix evidence variable G to +, and use 0.7 and 0.62 to determine that B is + and M is -. And use B and M and 0.24 to determine that I is +. Last we use G is + and 0.44 that J is +. Then we look for the table, with B,I,M are t,t,f, the weight is 0.8. After a lot of sample we could get a probability of the query using this method.

Naive Bayes Spam Classification

Implementation

a: Computing Features.

First we read all the training emails in and count the times the word occurs. If the time is larger than a value k we set, we put the word in a python data structure dictionary, with key equal the word and value equal to the times the word occur in the whole training set. We will use this data structure for later implementations too.

b: Training

In this part we calculate the likelihood of each word by dividing the occurance by the total word count in the lexicon, and save the likelihood to a new dictionary.

We smooth our likelihood by declaring a global variable m, and add m to the nominator and add m times the total word count in our lexicoin to the denominator. We try different m (1,5,10,20,50...) to see how the classfication changes and we discover that for ham testing the larger m is the more accurate the classfication is; for spam testing is the other way around.

We also estimate the prior P(class) by dividing the number of spam or ham documents by the sum of ham and spam documents respectvely.

c: testing

We classify the message by comparing the probability of being a ham and a spam. Since it is a binary classification we can use the following formula:

$$P(\text{spam} \mid \text{message}) > P(\neg\text{spam} \mid \text{message})$$

Since we have a estimate of prior P(class), our MAP estimation is better because we have a zero-one loss function while ML cannot give us.

Result

d:

After several test, we found that when k = 113, m = 50, the accuracy of spam and ham reach a balance (though both are not satisfying) spam accuracy: 22%, ham accuracy: 40%, overall accuracy: 31%

We experiment with k and m value a bit. m's influence is mentioned in b before. As for k, there's no propotional influence. Overall, as k increases, the accuracy of ham increases also except for some exceptional values. While it has no regular pattern for spam accuracy. We also noted that, when the k reaches 148 or higher, all emails will be classified as ham.

Example emails:

Subject: no risk kiosk pg lnbcer
hey
guy ,
yome
biot insted
sat it cr you
counterbalance
gtbmbbsoitsvn
pykn ifvreakjga
sdw ysenbzkjtw
ex bghcbgksga gp taokdl cmpnlbcg

When we have a garbled message we expect it is marked as spam while the classification failed. Our assumption is that meaningless words don't occur in spam learning set (even if they do they will not repeat in the test email) so we don't have a probability for these words.

Another example would be:
Subject: win a turbocharged jaguar xp in carter ' s casino affirmation
click
here
to transfer your free money into your account now .
f r e e m
o n e y !
this bonus is real and free by

downloading
our casino software and making your first purchase at the casino .
certain conditions apply .

When we have spaces between typical spam words our classification fails easily.

Interesting facts:

- We find that when k >= 148 all documents are indicated as ham.
- We have considered whether to include the punctuation as part of the word sequences. Because in spams there are more likely to be signs such as “$” and “%”

Member Contribution:

We regard all codes and report as the result of our equally hard work. We solved problem one together. On problem two and three we discuss about the algorithms together and write functions seperately, but test and debug together.