CS 188
Spring 2023

Introduction to
Artificial Intelligence

HW9 Part 2 Solutions

SP23 HW 9 Part 2 Solutions. [26 pts]

Q1) (3pts) Probably the most famous application of Naive Bayes classification is spam filtering. In this problem, we will assume that for each email there is a binary variable Y that takes the values *spam* or *ham* depending on whether the email is spam or not respectively. For each email, assume that each word w in the email follows a probability distribution $P(W = w|Y)$, and the variable W can take on words from some previously determined dictionary (note that the ordering of words in the email does not affect this probability). Punctuation in the email is ignored. For example: for an email with three words $w_1 = I$, $w_2 = love$ and $w_3 = CS188$, the label of the email is given by $\text{argmax}_y P(Y = y|w_1, w_2, w_3) = \text{argmax}_y P(Y = y) \prod_{i=1}^{3} P(W = w_i|Y = y)$.

Assume that we have trained a Naive Bayes classifier on a large dataset of emails using the above model. The following table has the estimated probabilities for some of the words.

W	Berkeley	is	amazing	Oski	rules
$P(W\|Y = \text{spam})$	1/6	1/8	1/4	1/4	1/8
$P(W\|Y = \text{ham})$	1/8	1/3	1/4	1/12	1/12

Table 1: Probability table.

Now assume that we receive a new two-word email that reads:

`Berkeley rules`

and we want to label it as spam or ham. Choose all the options for the value of $P(Y = spam)$ for which this new email will be classified as "spam" given the table above.

A. 0.0
B. 0.2
C. 0.4
D. 0.6
E. 0.8
F. 1.0

C,D,E,F.

$P(Y = spam|w_1 = Berkeley, w_2 = rules) > P(Y = ham|w_1 = Berkeley, w_2 = rules) \Rightarrow$
$P(Y = spam)P(w_1 = Berkeley|Y = spam)P(w_2 = rules|Y = spam) > P(Y = ham)P(w_1 = Berkeley|Y = ham)P(w_2 = rules|Y = ham) \Rightarrow$
$1/8 * 1/6 * P(Y = spam) > 1/12 * 1/8 * (1 - P(Y = spam)) \Rightarrow$
$P(Y = spam) > 1/3.$

Q2) (4pts) Now let's assume that we observe the following three emails with their true label in parentheses:

(Spam): `This is a warning that your social security number has been stolen.`
(Ham): `More cat and dog photos?`
(Ham): `I love exam prep, regular sections and social events.`

What are the estimates of the following probabilities using the Naive Bayes model?

Q2.1) $P(W = \text{warning}\|Y = \text{spam})$	A. 0	B. 1/10	C. 1/5	D. 1/3	E. 2/3	F. None of the options
Q2.2) $P(W = \text{social}\|Y = \text{ham})$	A. 0	B. 1/10	C. 1/5	D. 1/3	E. 2/3	F. None of the options
Q2.3) $P(W = \text{office}\|Y = \text{ham})$	A. 0	B. 1/10	C. 1/5	D. 1/3	E. 2/3	F. None of the options
Q2.4) $P(Y = \text{ham})$	A. 0	B. 1/10	C. 1/5	D. 1/3	E. 2/3	F. None of the options

(In order) F,F,A,E. We estimate the conditional probability of a word given the email label using word counts as follows:

$$P(W = \text{warning}|Y = \text{spam}) = \frac{c_w(W=warning,Y=spam)/c_w(total)}{c_w(spam)/c_w(total)} = \frac{c_w(W=warning,Y=spam)}{c_w(spam)} = \frac{1}{12}$$

where for instance $c_w(W = \text{warning}, Y = \text{spam})$ is equal to the number of occurrences of the word warning in a spam email and $c_w(\text{total})$ is the total number of words in spam emails. Similarly, $P(W = \text{social}|Y = \text{ham}) = 1/14$. The word office does not appear in the training emails and since we are not using smoothing the probability is $P(W = \text{office}|Y = \text{ham}) = 0$. Finally, $P(Y = ham) = c_e(Y = ham)/c_e(total) = 2/3$, where $c_e(Y = ham)$ and $c_e(total)$ are the number of ham emails and the total number of emails, respectively.

Q3) (3pts) Using the same dataset as in the previous question, we will now utilize the power of Laplace Smoothing with $k = 2$ in our classification task. Assume that the number of different words in our dictionary is V. Write expressions for the following probability estimates. Your expressions can depend on V and you should perform Laplace smoothing on the prior probabilities as well.

Q3.1) $P(W = \text{warning}|Y = \text{spam})$

"Warning" occurs once in the spam emails, and there are 12 total words in all spam emails. Smoothing with $k = 2$ gives $\frac{1+2}{12+2V}$.

Q3.2) $P(W = \text{social}|Y = \text{ham})$

"Social" occurs once in the ham emails, and there are 14 total words in all ham emails. Smoothing with $k = 2$ gives $\frac{1+2}{14+2V}$.

Q3.3) $P(Y = \text{ham})$

Same as before: number of ham emails divided by the number of spam emails using Laplace smoothing, which is $4/7$.

Q4) (3pts) We observe the following values for the accuracies on the test dataset for different values of k:

k	0	1	2	10
accuracy	0.65	0.68	0.74	0.6

Table 2: Accuracy and k values.

If for $k = 0$ we achieve an accuracy of 0.8 on the training set, then which of the following options are plausible accuracies for $k = 10$ also on the training set? Select all that apply.

A. 0.1

B. 0.7

C. 0.99

D. None of the above.

B.

On the training set, smoothing counts is not likely to help as much as on the test set. Smoothing is supposed to prevent "overfitting" – but overfitting is what gives really high accuracy on the training set. There is also an optimal spot for smoothing parameter values, and we see that at $k = 10$, we went past that optimal spot on the test set. So we have a technique that is designed to prevent overfitting (in other words, lower accuracy) on the training set, and it's so strong that it's actually hurting performance even on the test set: it definitely shouldn't be expected to increase performance on the training set! This rules out 0.99 as an answer.

The answer 0.1 can be ruled out too. The reason is that we observe a test accuracy of 0.6 for $k = 10$ so we would expect the accuracy in the training set to be similar to that and most likely higher for $k = 10$.

0.7, however, is a reasonable answer: it hurts performance compared to not doing smoothing, which is what we should expect.

Q5) The "Naive" assumption in Naive Bayes is quite strong. In this question we will try to improve the representational capacity of the classifier. More specifically, instead of assuming that the presence of each word depends only on the label via $P(W = w_i|Y)$, we will assume that this probability depends also on the previous word. i.e. $P(W_i = w_i|Y, W_{i-1})$. The model can also be seen in the following figure:

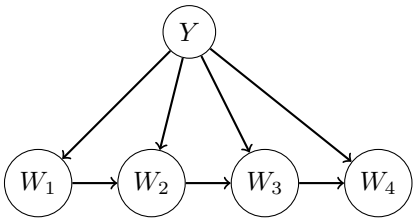

Figure 1: New Bayes Model.

Q5.1) (2pts) If our dictionary has V words, then which is the **minimal** number of conditional **word** probabilities that we need to estimate for this model?

A. V

B. $2V$

C. V^2

D. $2V^2$

E. 2^V

F. 2^{2V}

D. We need to consider V possible words for each one of V possible previous words and 2 possible labels: $2V^2$.

Q5.2) (2pts) If we use a large dataset with relatively equal number of spam and ham examples to train both this model and the original Naive model, then which of the following effects are expected to be true?

A. The entropy of the posterior $P(Y|W)$ will on average be lower in the new model.

B. The accuracy on the training data will be higher with the new model.

C. The accuracy on the test data will be higher with the new model.

D. None of the above.

B,C. The new model is closer to an actual model of language, and so should better model emails and thus filter spam from non-spam on both the training and test datasets.

This is the reasoning why the first option is incorrect. Remember that Naive Bayes is an overconfident model: it gives unwarrantedly low-entropy posteriors. This model is less "naıve", and is therefore less overconfident - its posterior can be expected to be higher entropy. As mentioned on Piazza you did not have to consider this option in your answers and even if you included it (correctly or not) it will not affect negatively your grade for this question.

Q6) (3pts) We now leave Naive Bayes and move to the perceptron algorithm.

Assume we want to train a multiclass perceptron with three classes A, B and C and with the initial values of the weights being $w_A = [1, 2]$, $w_B = [2, 0]$ and $w_C = [2, -1]$. We train the model on the following dataset:

x_0	x_1	label
1	1	A

Table 3: Training dataset.

What are the values of the vectors w_A, w_B and w_C after training using the data sample above only once?

Q6.1) $w_A = [1, 2]$

Q6.2) $w_B = [2, 0]$

Q6.3) $w_C = [2, -1]$

The predicted label is $y' = \text{argmax}_{y \in \{A,B,C\}} w_y \cdot [x_0, x_1] = A$, since $w_A \cdot [x_0, x_1] = [1,2] \cdot [1,1] = 3$ is greater than both $w_B \cdot [x_0, x_1] = [2,0] \cdot [1,1] = 2$ and $w_C \cdot [x_0, x_1] = [2,-1] \cdot [1,1] = 1$. Since the predicted label $y' = A$ is equal to the correct label $y^* = A$, the weights are not updated, so the weights are the same as the initial ones.

Q7) (3pts) Now assume that we have a different multiclass perceptron model that we want to train a multiclass perceptron with three classes A, B and C and with the initial values of the weights being $w_A = [2,4]$, $w_B = [-1,0]$ and $w_C = [2,-2]$. We train the model on the following dataset:

x_0	x_1	label
-2	1	C

Table 4: Training dataset.

What are the values of the vectors w_A, w_B and w_C after training using the data sample above only once?

Q7.1) $w_A = [2,4]$
Q7.2) $w_B = [1,-1]$
Q7.3) $w_C = [0,-1]$

The predicted label is $y' = \arg\max_{y \in \{A,B,C\}} w_y \cdot [x_0, x_1] = B$, since $w_B \cdot [x_0, x_1] = [-1,0] \cdot [-2,1] = 2$ is greater than both $w_A \cdot [x_0, x_1] = [2,4] \cdot [-2,1] = 0$ and $w_C \cdot [x_0, x_1] = [2,-2] \cdot [-2,1] = -6$. Since the predicted label y' = B is not equal to the correct label $y^* = C$, the weights are updated by subtracting the datum from the weights of the predicted label and by adding the datum to the weights of the correct label:

$$w_B \leftarrow w_B - [-2,1] = [1,-1]$$
$$w_C \leftarrow w_C + [-2,1] = [0,-1]$$

Q8) (3pts) Now assume that we have a different multiclass perceptron model that we want to train a multiclass perceptron with three classes A, B and C and with the initial values of the weights being $w_A = [1,0]$, $w_B = [1,1]$ and $w_C = [3,0]$. We train the model by iterating infinitely over the following dataset:

training sample i	x_0	x_1	label
0	1	1	A
1	-1	1	B
2	1	-1	C
3	-1	-1	A

Table 5: Training dataset.

Convergence means that the estimated values do not change anymore by passing through the dataset. Which of the following options must be **true** without any additional information?

A. All weight vectors w_A, w_B, w_C converge.

B. Only two of the weight vectors w_A, w_B, w_C converge.

C. Only one of the weight vectors w_A, w_B, w_C converge.

D. None of the weight vectors w_A, w_B, w_C converge.

E. None of the above.

C. First, notice that the data is not linearly separable, so not all of the weight vectors converge (this is the case for the perceptron). Second, notice that it's impossible for only two of the weight vectors to converge since every time the prediction is wrong, two of the weights are updated with a datum (this is the case for the multiclass perceptron), all of which are non-zero for this particular data set. Therefore, there are two possibilities left: either only one of the weight vectors converge or none of them converge. To find out which one is the case, you can do a single pass through the data in order to notice a pattern emerge.

For the training example with $i = 0$, the perceptron incorrectly predicts a label of C, so the weight vectors w_A and w_C get updated to $w_A = [2, 1]$ and $w_C = [2, -1]$. Then, for the training examples with $i = 1$ and $i = 2$, the perceptron correctly predicts the labels B and C, respectively, so the weight vectors are not updated. Then, for the training example with $i = 3$, the perceptron incorrectly predicts a label of C, so the weight vectors w_A and w_C updated to $w_A = [1, 0]$ and $w_C = [3, 0]$, which are the values that you originally started with! Therefore, as you train the perceptron on the data set an infinite number of times, the value of the weight vector w_A fluctuates between $w_A = [2, 1]$ and $w_A = [1, 0]$, while the value of the weight vector w_C fluctuates between $w_C = [2, -1]$ and $w_C = [3, 0]$. Meanwhile, the value of the weight vector w_B stays at $w_B = [1, 1]$, and therefore it is the only weight vector that converges.