CS 188 Spring 2022 Introduction to Artificial Intelligence — Written HW 9

Due: Wednesday 04/13/2022 at 10:59pm (submit via Gradescope).

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: It is recommended that your submission be a PDF that matches this template. You may also fill out this template digitally (e.g. using a tablet). **However, if you do not use this template, you will still need to write down the below four fields on the first page of your submission.**

First name	Qingjing
Last name	Zhang
SID	3037581096
Collaborators	none.

For staff use only:

Q1.	Probabilistic Language Modeling	/40
	Total	/40

Q1. [40 pts] Probabilistic Language Modeling

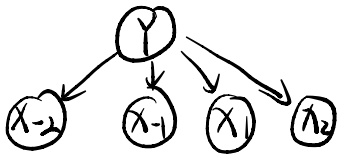
In lecture, you saw an example of supervised learning where we used Naive Bayes for a binary classification problem: to predict whether an email was ham or spam. To do so, we needed a labeled (i.e., ham or spam) dataset of emails. To avoid this requirement for labeled datasets, let's instead explore the area of unsupervised learning, where we don't need a labeled dataset. In this problem, let's consider the setting of language modeling.

Language modeling is a field of Natural Language Processing (NLP) that tries to model the probability of the next word, given the previous words. Here, instead of predicting a binary label of "yes" or "no," we instead need to predict a multiclass label, where the label is the word (from all possible words of the vocabulary) that is the correct word for the blank that we want to fill in.

One possible way to model this problem is with Naive Bayes. Recall that in Naive Bayes, the features $X_1, ..., X_m$ are assumed to be pairwise independent when given the label Y. For this problem, let Y be the word we are trying to predict, and our features be X_i for $i = -n, ..., -1, 1, ..., n$, where $X_i = ith$ word i places from Y. (For example, X_{-2} would be the word 2 places in front of Y. Again, recall that we assume each feature X_i to be independent of each other, given the word Y. For example, in the sequence `Neural networks ____ a lot`, X_{-2} = `Neural`, X_{-1} = `networks`, Y = the blank word (our label), X_1 = `a`, and X_2 = `lot`.

(a) First, let's examine the problem of language modeling with Naive Bayes.

(i) [1 pt] Draw the Bayes Net structure for the Naive Bayes formulation of modeling the middle word of a sequence given two preceding words and two succeeding words. You may think of the example sequence listed above:
`Neural networks ____ a lot.`



(ii) [1 pt] Write the joint probability $P(X_{-2}, X_{-1}, Y, X_1, X_2)$ in terms of the relevant Conditional Probability Tables (CPTs) that describe the Bayes Net.

$$P(X_{-2}, X_{-1}, Y, X_1, X_2) = P(X_{-2}|Y) \cdot P(X_{-1}|Y) P(X_1|Y) P(X_2|Y) P(Y)$$

(iii) [1 pt] What is the size of the largest CPT in the Bayes net? Assume a vocabulary size of V, so each variable can take on one of possible V values.

V^2.

(iv) [1 pt] Write an expression of what label y that Naive Bayes would predict for Y (Hint: Your answer should involve some kind of arg max and CPTs.)

$$\arg\max_y \; P(X_{-2}|Y) \cdot P(X_{-1}|Y) P(X_1|Y) P(X_2|Y) P(Y)$$

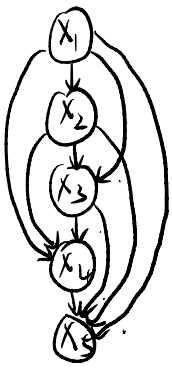
(v) [3 pts] Describe 2 problems with the Naive Bayes Approach for the general problem of language modeling. Hint: do you see any problems with the assumptions that this approach makes?

① feature X_i isn't usually independent pairwise given Y, since words are usually related to each other.
② Y may related to any feature X_i
the order of X_i may matter.

Now, let's change our setting a bit. Instead of trying to fill in a blank given surrounding words, we are now only given the preceding words. Say that we have a sequence of words: $X_1, ..., X_{m-1}, X_m$. We know $\{X_i\}_{i=0}^{m-1}$ but we don't know X_m.

(b) For this part, assume that every word is conditioned on all previous words. We will call this the **Sequence Model**.

(i) [1 pt] Draw the Bayes Net (of only X_1, X_2, X_3, X_4, X_5) for a 5-word sequence, where we want to predict the fifth word in a sequence X_5 given the previous 4 words X_1, X_2, X_3, X_4. Again, we are assuming here that each word depends on all previous words.



(ii) [1 pt] Write an expression for the joint distribution of a general sequence of length m: $P(X_1, ..., X_m)$.

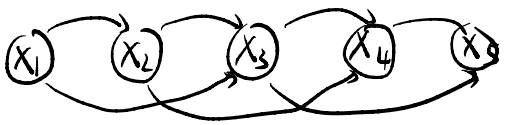
$P(X_1, \cdots X_m) = P(X_m \mid X_{1:m-1}) \cdot P(X_{m-1} \mid X_{1:m-2}) \cdots P(X_2 \mid X_1) \cdot P(X_1)$

(iii) [1 pt] What is the size of the largest CPT in the Bayes net? Assume a vocabulary size of V, so each variable can take on one of possible V values.

V^5

(c) You should have gotten a very large number for the previous part, which shows how infeasible the sequence model is. Instead of the model above, let's now examing another modeling option: N-grams. In N-gram language modeling, we add back some conditional assumptions to bound the size of the CPTs that we consider. We limit the tokens of consideration from "all previous words" to instead using only "the previous $N-1$ words." This creates the conditional assumption that, given the previous $N-1$ words, the current word is independent of any word before the previous $N-1$ words. For example, for $N = 3$, if we are trying to predict the 100th word, then given the previous $N - 1 = 2$ words (98th and 99th words), then the 100th word is independent of words $1, \ldots, 97$ of the sequence.

(i) [1 pt] Making these additional conditional independence assumption changes our Bayes Net. Redraw the Bayes Net from part (ci) to represent this new N-gram modeling of our 5-word sequence: X_1, X_2, X_3, X_4, X_5. Use $N = 3$.



$P(X_5 \mid X_4, X_3) \; P(X_4 \mid X_3, X_2) \; P(X_3 \mid X_2, X_1) \; P(X_2 \mid X_1) \; P(X_1)$

N-word as group

(ii) [2 pts] Write an expression for the N-gram representation of the joint distribution of a general sequence of length m: $P(X_1, ..., X_m) = P(X_{1:m})$. Your answer should express the joint distribution $P(X_{1:m})$ in terms of m and N.

Hint: If you find it helpful, try it for the 5 word graph above first before going to a general m length sequence.

$$P(X_{1:m}) = \left[\prod_{i=N}^{m} P(X_i \mid X_{i-N:i-1})\right]\left[\prod_{i=1}^{N-1} P(X_i \mid X_{1:i-1})\right] \cdot P(X_1)$$

(iii) [1 pt] What is the size of the largest CPT in the Bayes net above? Again, assume a vocabulary size of V, and $m > N$.

V^N

(iv) [2 pts] Describe one disadvantage of using N-gram over Naive Bayes.

Since the CPT size is larger, it takes longer time to compute.

(v) [4 pts] Describe an advantage and disadvantage of using N-gram over the Sequence Model above.

pro: it combine some information happen before

con: the result is affected by data sparsity

(d) In this question, we see a real-world application of smoothing in the context of language modeling.

Say we have the following training corpus from Ted Geisel:

`i am sam . sam i am . i do not like green eggs and ham .`

Consider the counts given in the tables below, as calculated from the sentence above.

1-gram	
Token	Count
i	3
am	2
sam	2
.	3
do	1
not	1
like	1
green	1
eggs	1
and	1
ham	1
TOTAL	17

2-gram phrases starting with i		
Token1	Token2	Count
i	am	2
i	do	1
TOTAL		3

2-gram phrases starting with am		
Token1	Token2	Count
am	sam	1
am	.	1
TOTAL		2

(i) [1 pt] Based on the above dataset and counts, what is the N-gram estimate for $N = 1$, for the sequence of 3 tokens `i am ham`? In other words, what is $P(i, am, ham)$ for $N = 1$?

$P(i, am, sam) = \frac{3}{17} \times \frac{2}{17} \times \frac{1}{17} = \frac{6}{4913} \approx 0.0012$

(ii) [1 pt] Based on the above dataset and counts, what is the N-gram estimate for $N = 2$, for the sequence of 3 tokens `i am ham`? In other words, what is $P(i, am, ham)$ for $N = 2$?

$P(i, am, ham) = 0$

(iii) [5 pts] Perform Laplace k-smoothing on the above problem and re-compute $P(i, am, ham)$ with the smoothed distribution, for $N = 2$. In order to calculate this, complete the pseudocount column for each entry in the probability tables. Note we add a new `<unk>` entry, which represents any token not in the table.

Hint: the count for the new `<unk>` row in each table would be 0.

1-gram		
Token	Count	Pseudocount
i	3	5
am	2	4
sam	2	4
.	3	5
do	1	3
not	1	3
like	1	3
green	1	3
eggs	1	3
and	1	3
ham	1	3
<unk>	0	2.
TOTAL	17	41

2-gram phrases starting with "i"			
Token1	Token2	Count	Pseudocount
i	am	2	4
i	do	1	3
i	<unk>	0	2.
TOTAL		3	9

2-gram phrases starting with "am"			
Token1	Token2	Count	Pseudocount
am	sam	1	3
am	.	1	3
am	<unk>	0	2.
TOTAL		2	8.

$P(i, am, ham) = \frac{4}{9} \times \frac{2}{8} \times \frac{5}{41} \approx 0.036$

(iv) [3 pts] What is the importance of smoothing in the context of language modeling?

Hint: see your answer for the previous subquestion.

Smoothing can tackle the problem of data sparsity ie. $f(w_i | w_{i-1}) = 0$

(v) [4 pts] What is a potential problem with Laplace smoothing? Propose a solution. (Assume that you have chosen the best k, so finding the best k is not a problem.)
Hint: Consider the effect of smoothing on a small CPT.

when training on small dataset,
effort of laplace smoothing on poor i probability is large.

Solution: add more data will diminish the effort of laplace smoothing.

(vi) [2 pts] Let the likelihood $\mathcal{L}(k) = P(i, am, sam)$. Give an expression for the log likelihood, $\ln \mathcal{L}(k)$, of this sequence after k-smoothing. Continue to assume $N = 2$.

$$P(i, am, sam) = P(am|i)\ P(sam|am) \cdot P(i).$$

$$\ell(k) = \log P(am|i) + \log P(sam|am) + \log P(i)$$

(vii) [4 pts] Describe a procedure we could do to find a reasonable value of k. No mathematical computations needed.
Hint: you might want to maximize the log likelihood $\ln \mathcal{L}(k)$ on something.

```
for R = [0, ..., N]
    if ℓ(R) > max_likelihood:
        best_R = R,
        max_likelihood = ℓ(R).
```