Homework 1

BT3102 - Computational Methods for Business Analytics
Due: 30-01-2022 (11:59 PM)

Question 1. NUS would like to investigate certain departments for gender biases in their PhD program admissions. There are two departments, dept#1 and dept#2, to which students can apply.(Dept is a random variable with values dept#1 and dept#2 in its domain.) Students can only apply to one, but not both. Students have a gender (male or female), and are either admitted or not. The table below gives the percent of students in each category.

Dept	Gender	Admitted	Percent
dept#1	male	true	30
dept#1	male	false	18
dept#1	female	true	8
dept#1	female	false	3
dept#2	male	true	5
dept#2	male	false	14
dept#2	female	true	8
dept#2	female	false	14

(a) What is $P(Admitted = true|Gender = male)$?
What is $P(Admitted = true|Gender = female)$?
Which gender is more likely to be admitted? (1 point)

(b) What is $P(Admitted = true|Gender = male; Dept = dept\#1)$?
What is $P(Admitted = true|Gender = female; Dept = dept\#1)$?
Which gender is more likely to be admitted to dept#1? (1 point)

(c) What is $P(Admitted = true|Gender = male; Dept = dept\#2)$?
What is $P(Admitted = true|Gender = female; Dept = dept\#2)$?
Which gender is more likely to be admitted to dept#2? (1 point)

(d) This is an instance of Simpson's paradox. Why is it a paradox? Explain why it happened in this case. (3 points)

Question 2. After your annual medical examination, the doctor gave you bad news and good news. The bad news is that you tested positive for a serious disease, and the test is 99% accurate (i.e., the probability of testing positive when you do have the disease is 0.99, as is the probability of testing negative when you do not have the disease). The good news is that this is a rare disease, striking only 1 in 10,000 people of your age.

(a) What is the probability that you actually have the disease given your test result? (2 points)

(b) Why is it good news that the disease is rare? (1 point)

Question 3. Independence between random variables A and B can be written as

(a) $P(A|B) = P(A)$,

(b) $P(B|A) = P(B)$, and

(c) $P(A, B) = P(A)P(B)$.

Prove that all three forms are equivalent. (4 Points)

Question 4 (7 points). You may have observed several spelling correction models Google search for instance. In this question, you will write a spelling corrector in Python 3 using Bayes' rule. Download spellcorrector.py, words.txt, testwords.txt from Piazza.

To find the correct spelling of a misspelled word w, we can generate multiple candidates of the correct word, and evaluate each candidate c using Bayes' rule.

$$P(c|w) = \frac{P(w|c)P(c)}{P(w)}$$

What we want is the candidate that maximizes $P(c|w)$, i.e.,

$$\text{argmax}_{c \in Candidates} \frac{P(w|c)P(c)}{P(w)}$$

Observe that $P(w)$ is constant for all candidates, so we can ignore it and simply fin the c that maximizes $P(w|c)P(c)$. Add code to the probability_of_word_given_candidate() method in spellcorrector.py.

The code in spellcorrector.py first reads from a large text corpus words.txt to obtain the number of occurrences of each word in the corpus. Then it reads misspelled words from testwords.txt. For each misspelled word w, it generates multiple candidates that are a 'distance' of one or two away from the misspelled word in the hope that the correct word is among them. Intuitively, the larger the distance between w and a candidate c, the greater the difference between them. A correct candidate should not be too far from w. Next, the code calls the probability of word given candidate() method to compute $P(w|c)P(c)$ for each candidate and pick the best candidate. It is your job to argure out how to compute each of the terms in $P(w|c)P(c)$. You may wish to peruse the entire spellcorrector.py file to understand the full flow of the code.

The code will print out the words that are wrongly corrected. Your code should correct at least 50% of the misspelled test words (please do not hard code to get exactly and only the words in testwords.txt correct). At the top of your code, indicate the fraction of test words you got correct.

Practice Questions

These questions need not be submitted. They are provided for your practice to help you prepare for midterm or final, especially to let you think deep and obtain better understanding of the material.

Question 1. In a letter dated 24 August 1654, Pascal labored to determine how a pot of money should be divided when a gambling game had to be terminated prematurely. Both Pascal and Fermat made several erroneous stabs at the problem before solving it. Now it is your turn. You are, of course, expected to provide the correct answer. This question will also help you exercise your possibly-atrophied-but-hopefully-still-intact recursion muscle.

In a game where each turn is made up of the roll of a die, player E gets one point when the die is even, and player O gets one point when the die is odd. The first player to accumulate 7 points wins the pot. Suppose the die is fair and the game is interrupted with E leading 4—2

(a) Denote the probability of the die being even as p, the winning number of points as m (7 in the above description), E's current points as e (e.g., 4 above), and O's current points as o (e.g., 2 above). In terms of p, m, e and o, what is the general recursive formula that fairly divides the money between E and O?

(b) Write Python 3 code to evaluate the recursive formula to determine the fraction of money that E gets in the above situation where p = 0:5, e = 4, o = 2 and m = 7. Write down E's proportion.

(c) What is the general non-recursive formula that fairly divides the money between E and O in terms of p, m, e and o?