CSE 486/586, Assignment 3

Joshua Crail (Crailjc)

Due: Tuesday, October 26, 2021, by 11:59 pm.

Note 1: The total mark for this assignment is 30. You should *NOT* directly copy anything from slides or other resources. You may get the ideas from slides but what you submit *must be in your own words*. Any help must be acknowledged.

Note 2: Question 6 is only for CSE 586 students (0 point for correct answer, and -5 for wrong answer)

1. Implement the min-conflicts for the 8-queens problem (assume that there is exactly one queen in each column and each row). (10 points)

   See attached code.

2. Explain in detail how can we construct a general and powerful spam filter using Naive Bayes Classifiers. Write your solution mathematically as discussed in class. (8 points)

   We start with a collection of labeled emails (either spam or ham). We iterate over the labelled spam emails and for every word (w) in that collection we, count how many spam emails contain that word w $P(w \mid S) = \frac{|spam\ emails\ containing\ w|+1}{|spam\ emails|+2}$ and how many times that word w is found in ham emails $P(w \mid H) = \frac{|spam\ emails\ containing\ w|+1}{|ham\ emails|+2}$
   Once all labeled emails have been iterated through and all data collected compute the percentage of spam emails from total data set $P(S) = \frac{|spam\ emails|}{|spam\ emails|+|ham\ emails|}$ and the percentage of ham emails from total data set $P(S) = \frac{|ham\ emails|}{|spam\ emails|+|ham\ emails|}$
   After this the initial training has been done and now given a unlabeled test email it can be determined if its spam or ham. From the unlabeled test email create a set $\{x_1, \dots, x_n\}$ of district words that you have seen before. Words that were not in the training emails are ignored. Then compute the probability of this email being spam given all the words in our set (and data associated with each word from our training)

   $$P(S \mid x_1, \dots, x_n) \approx \frac{P(S)\prod_{i=1}^{n} P(x_i \mid S)}{P(S)\prod_{i=1}^{n} P(x_i \mid S) + P(H)\prod_{i=1}^{n} P(x_i \mid H)}$$

   Once this has been calculated if the result of $P(S \mid x_1, \dots, x_n)$ is greater than 0.5 then the output will be spam (spam email) else it will be ham (good email). The reason for the plus one in the numerator is in case someone puts a word in a spam email and our data set has only ever seen that word in ham emails meaning that $P(w \mid S)$ would be 0 and our calculation of $P(S \mid x_1, \dots, x_n)$ would become zero and the email would be considered ham even though its actually spam. The plus one is there to prevent people

from getting around our filter by using a word that has only been seen in ham emails or the reverse where a person accidently puts a spam word in a ham email.
The denominator is plus two because $|spam\ emails| + 2$ =
$|spam\ emails\ containing\ w| + 1\ + |spam\ emails\ \ not\ containing\ w| + 1$

3. Implement the above spam filter. (Optional group problem, 3 EXTRA credits, Due: November 10, 2021, show me your code in person)

4. After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease and that the test is 99% accurate. The good news is that this is a rare disease, striking only 1 in 10,000 people of your age. What are the chances that you actually have the disease? (8 points)

   D+ = Have disease D- = Do not have disease T+ = Test positive T- = Test negatice
   Find $P(D \mid T)$

   $$P(D+ \mid T+) = \frac{P(T+ \mid D+)\,P(D+)}{P(T+)}$$

   $$P(D+) = \frac{1}{10{,}000}\ P(D-) = \frac{9999}{10{,}000}\ \ P(T+ \mid D+) = \frac{99}{100}$$
   $$P(T- \mid D-) = \frac{99}{100}\ P(T+ \mid D-) = \frac{1}{100}$$
   $$P(T+) = \ P(T+|D+)\,P(D+) + P(T+|D-)P(D-) = 0.010098$$

   $$P(D+ \mid T+) = \frac{P(T+ \mid D+)\,P(D+)}{P(T+)} = \frac{.99 * 0.0001}{0.010098} = 0.0098$$

5. Which algorithm discussed in class has been used to schedule observations for the Hubble Space Telescope, reducing the time taken to schedule a week of observations from three weeks (!) to around 10 minutes? What is the name of a well-known CSP solver we described in class? (4 points)

   The Constraint Satisfaction Problem (CSP) algorithm. The well-known CSP solver is Google OR-Tools

6. We wish to transmit an *n*-bit message to a receiving agent. The bits in the message are independently corrupted (flipped) during transmission with ε probability each. With an extra parity bit sent along with the original information, a message can be corrected by the receiver if at most one bit in the entire message (including the parity bit) has been corrupted. Suppose we want to ensure that the correct message is received with probability at least 1−δ. What is the maximum feasible value of *n*? (Only CSE 586 students)

   Message can only be fixed when 0 or 1 bits are corrupted any more than that results in a message that cannot be corrected. As such we need to find the probability of either a single bit being corrupted or no bit corruption to determine the max feasibility value of n.

   The probability of zero corruption is $(1 - \varepsilon)^{n+1}$
   The probability of one corruption is $(n + 1)\ \varepsilon\ (\ 1 - \varepsilon\ )^{n}$

   Thus $(n + 1)\ \varepsilon\ (\ 1 - \varepsilon\ )^{n} + (1 - \varepsilon)^{n+1} \geq (1 - \delta)$ where the solution to the inequality will result in the max value of n that satisfies the probability of at least $(1 - \delta)$.