Collab Worksheet 3 Solutions

CS440/ECE448, Spring 2021

Week of 2/19 - 2/24, 2021

Question 1

You're on a phone call with your friend, trying to help figure out why their computer won't start. There are only two possibilities, $Y = \text{CPU}$, or $Y = \text{PowerSupply}$, with prior probability $P(Y = \text{CPU}) = 0.3$.

You ask your friend whether the computer makes noise when they try to turn it on. There are two possibilities, $E = \text{quiet}$, and $E = \text{loud}$. You know that a power supply problem often leaves a quiet computer, but that the relationship is stochastic, as shown:

$$P(E = \text{noise}|Y = \text{CPU}) = 0.8, \quad P(E = \text{noise}|Y = \text{PowerSupply}) = 0.4$$

(a) What is the MAP classifier function $\hat{Y}(E)$, as a function of E?

> **Solution:** The joint probabilities of evidence and label are:
>
> $$P(\text{noise}, \text{CPU}) = 0.24, \quad P(\text{noise}, \text{PowerSupply}) = 0.28$$
> $$P(\text{quiet}, \text{CPU}) = 0.06, \quad P(\text{quiet}, \text{PowerSupply}) = 0.42$$
>
> Choosing the maximum *a posteriori* label given each observation gives
>
> $$\hat{Y}(\text{noise}) = \text{PowerSupply}, \quad \hat{Y}(\text{quiet}) = \text{PowerSupply}$$
>
> In other words, regardless of whether the computer is noisy or quiet, the power supply is always the most probable source of the problem.

(b) What is the Bayes error rate?

> **Solution:** The Bayes error rate is the probability of error of the optimal classifier, which is $P(\text{noise}, \text{CPU}) + P(\text{quiet}, \text{CPU}) = 0.3$.

(c) CPU damage is more expensive than power supply damage, so let's define a false alarm to be the case where your classifier says $\hat{Y} = \text{CPU}$, but the actual problem is $Y = \text{PowerSupply}$. Under this definition, what are the false-alarm rate and missed-detection rate of the MAP classifier?

Solution: The MAP classifier always guesses "Power Supply," so the false alarm and missed detection rates are

$$P(\hat{Y} = \text{CPU}|Y = \text{PowerSupply}) = 0.0$$
$$P(\hat{Y} = \text{PowerSupply}|Y = \text{CPU}) = 1.0$$

Question 2

Consider the following binary logic function:

$$y = \neg\left((x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3)\right)$$

Convert truth values to numbers in the obvious way: let $x_i = 1$ be a synonoym for $x_i =$ **True**, and let $x_i = 0$ by a synonym for $x_i =$ **False**. Let $x = [x_1, x_2, x_3]^T$ and $w = [w_1, w_2, w_3]^T$, let $x^T w$ denote the dot product of vectors x and w, and let $u(\cdot)$ denote the unit step function. Find a set of parameters w_1, w_2, w_3 and b such that the logic function shown above can be computed as $y = u(w^T x + b)$.

Solution: Drawing up a truth table, we get

x_1	x_2	x_3	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Looking at the truth table (or simplifying the original formula), we see that $y = 1$ unless $x_1 = 1$ and $x_3 = 0$, e.g., unless $x_1 - x_3 > 0.5$. We can write this as, for example, $y = u\left(-x_1 + x_3 + 0.5\right)$, i.e., $w = [-1, 0, 1]$, $b = 0.5$.

Question 3

We want to implement a classifier that takes two input values, where each value is either 0, 1 or 2, and outputs a 1 if at least one of the two inputs has value 2; otherwise it outputs a 0. Can this function be implemented by a linear classifier? If so, construct a linear classifier that does it; if not, say why not.

Solution: In this case the input space of all possible examples with their target outputs is:

	0	1	2
2	1	1	1
1	0	0	1
0	0	0	1

Since there is clearly no line that can separate the two classes, this function is not linearly separable and so it cannot be learned by a Perceptron.

Question 4

Consider a problem with a binary label variable, Y, whose prior is $P(Y = 1) = 0.4$. Suppose that there are 100 binary evidence variables, $E = [E_1, \ldots, E_{100}]$, each with likelihoods given by $P(E_i = 1|Y = 0) = 0.3$ and $P(E_i = 1|Y = 1) = 0.8$ for $1 \leq i \leq 100$.

(a) Specify the classifier function, $\hat{y}(e)$, for a naive Bayes classifier, where $e = [e_1, \ldots, e_{100}]$ is the set of observed values of the evidence variables.

Solution:

$$\hat{y}(e) = \begin{cases} 1 & P(Y=1)\prod_{i:e_i=1} P(E_i = 1|Y = 1)\prod_{i:e_i=0} P(E_i = 0|Y = 1) > \\ & (1 - P(Y = 1))\prod_{i:e_i=1} P(E_i = 1|Y = 0)\prod_{i:e_i=0} P(E_i = 0|Y = 0) \\ 0 & \text{otherwise} \end{cases}$$

Plugging in the given parameter values, we have

$$\hat{y}(e) = \begin{cases} 1 & 0.4\prod_{i=1}^{100}(0.8)^{e_i}(0.2)^{1-e_i} > 0.6\prod_{i=1}^{100}(0.3)^{e_i}(0.7)^{1-e_i} \\ 0 & \text{otherwise} \end{cases}$$

(b) The naive Bayes classifier can be written as

$$\hat{y}(e) = \begin{cases} 1 & w^T e + b > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $w^T e$ is the dot product between the vectors w and e. Find w and b (write them as expressions in terms of constants; don't simplify).

Solution:

$$\hat{y}(e) = \begin{cases} 1 & \ln(0.4) + \sum_{i=1}^{100} e_i \ln(0.8) + (1 - e_i)\ln(0.2) > \ln(0.6) + \sum_{i=1}^{100} e_i \ln(0.3) + (1 - e_i)\ln(0.7) \\ 0 & \text{otherwise} \end{cases}$$

so the parameters are

$$\begin{aligned} b &= \ln(0.4) - \ln(0.6) + 100\ln(0.2) - 100\ln(0.7) \\ w_i &= \ln(0.8) - \ln(0.2) - \ln(0.3) + \ln(0.7), \quad 1 \leq i \leq 100 \end{aligned}$$