ISYE 6420: Bayesian Statistics **Spring 2020**

Homework #1

Instructor: Roshan Vengazhiyil, Brani Vidakovic *Name:* Nick Korbit, *gtID:* 903263968

Problem 1

a). We first calculate the sensitivity and specificity for the serial system. As per [1]:

$$\text{Se} = \text{Se}_1 \times \text{Se}_2 \times \cdots \times \text{Se}_k$$
$$\text{Sp} = 1 - [(1 - \text{Sp}_1) \times (1 - \text{Sp}_2) \times \cdots \times (1 - \text{Sp}_k)]$$

We now combine three tests – Tinel's sign (TS), Phalen's test (PH) and the nerve conduction velocity test (NCV). So that:

$$Se = 0.97 \times 0.92 \times 0.93 = 0.829932$$

$$Sp = 1 - (1 - 0.91) \times (1 - 0.88) \times (1 - 0.87) = 0.998596$$

b). We then turn to the parallel systems. As per [1]:

$$\text{Se} = 1 - [(1 - \text{Se}_1) \times (1 - \text{Se}_2) \times \cdots \times (1 - \text{Se}_k)]$$
$$\text{Sp} = \text{Sp}_1 \times \text{Sp}_2 \times \cdots \times \text{Sp}_k$$

We combine three tests:

$$Se = 1 - (1 - 0.97) \times (1 - 0.92) \times (1 - 0.93) = 0.999832$$

$$Sp = 0.91 \times 0.88 \times 0.87 = 0.696696$$

c). After finding Sp and Se we calculate PPV for both serial and parallel systems. As per [1]:

$$\text{PPV} = \frac{\text{Se} \times \text{Pre}}{\text{Se} \times \text{Pre} + (1 - \text{Sp}) \times (1 - \text{Pre})}$$

We first find Pre as $50/1000 = 0.05$. The we find

$$\text{PPV}_{\text{serial}} = \frac{0.829932 \times 0.05}{0.829932 \times 0.05 + (1 - 0.998596) \times (1 - 0.05)} \approx 0.96885857$$

And

$$\text{PPV}_{\text{parallel}} = \frac{0.999832 \times 0.05}{0.999832 \times 0.05 + (1 - 0.696696) \times (1 - 0.05)} \approx 0.14784710$$

Problem 2

For the second problem we are basically to build a Naive Bayes classifier. We start with calculating priors – $P(Class =' WentBeach') = 40/100 = 0.4$ and $P(Class = not' WentBeach') = 1 - 0.4 = 0.6$.

Then we go to the "train" phase. We calculate conditional probabilities for each feature – 'Midterm', 'Finances', 'Friends Go', 'Forecast' and 'Gender'. That's easy as all features have binary output, so we just calculate means for cases 'Went Beach'=True and 'Went Beach'=False. We cache those values.

In the the "predict" phase we multiply all conditional probabilities depending on each person parameters and then multiply again by the prior. We do that for both 'Went Beach'=True and 'Went Beach'=False classes. Then we normalize the output.

Let's test our classifier for three hypothetical persons – Jane, Michael and Melissa. We can parameterize each one with a dictionary:

```
    jane = {
'Midterm': 1,
'Finances': 1,
'Friends Go': 0,
'Forecast': 0,
'Gender': 1,
}
```

Then we run "predict" phase and analyze the output:

- Jane, True: 0.17238060388100107, False: 0.8276193961189989
- Michael, True: 0.4070417547103887, False: 0.5929582452896114
- Melissa, True: 0.2796420404874101, False: 0.7203579595125899

We see that Jane's output matches the HW1 example and the highest probability to go to the beach belongs to Michael - around 40%.

Note. Both code (hw1q2.py) and data (naive.csv) are included in the zip archive. The only code dependency is pandas package. To run the code just run *'python hw1q2.py'*. Tested for Python 3.8.

Problem 3

Let's first parameterize the problem. We introduce two binary variables – 'Knowledge' (K) and 'Question' (Q). Knowledge is set to True if a student knows the answer to the question, otherwise it's False. Question is True if an answer to the given question is correct, otherwise it's False. We can represent the system via causality $K \to Q$, so that Q depends from K. We are also given the probabilities: $P(K) = 0.8$, $P(Q|K) = 1.0$ and $P(Q|\neg K) = 0.25$.

a)+c). As the questions are independent let's first find the probability of one question to be answered correctly or, more formally, let's find the total probability $P(Q)$:

$$P(Q) = P(Q|K) \times P(K) + P(Q|\neg K) \times P(\neg K)$$

So that

$$P(Q) = 1.0 \times 0.8 + 0.25 \times 0.2 = 0.85$$

If we have n independent questions then the probability of *all* the questions to be answered correctly will be calculated in the serial manner: $P_n = P(Q)^n$. If $P(Q) < 1$ then with $n \to \infty$ we have $P_n \to 0$.

In the case of $n = 2$, we have $P_2 = 0.85^2 = 0.7225$.

b)+c). Let's then find a probability of Knowledge if a question was answered correctly – $P(K|Q)$:

$$P(K|Q) = \frac{P(Q|K) \times P(K)}{P(Q)}$$

We have already calculated $P(Q)$, so let's find $P(K|Q)$:

$$P(K|Q) = \frac{1.0 \times 0.8}{0.85} \approx 0.94117647$$

If we have n independent questions then the probability of having known *all* the questions will be calculated in the serial manner: $P_n = P(K|Q)^n$. If $P(K|Q) < 1$ then with $n \to \infty$ we have $P_n \to 0$.

In the case of $n = 2$, we have $P_2 = 0.94117647^2 \approx 0.88581315$.

References

[1] Engineering Biostatistics: An Introduction using MATLAB and WinBUGS. Brani Vidakovic - Wiley Series in Probability and Statistics.