COMP307/AIML420 – Fundamentals of AI
Assignment 3: Reasoning under uncertainty

Part 1: Naïve Bayes Method

This report will be looking at the Naïve Bayes classifier which I implemented. This specific model looks at the ‘breast-cancer-training.csv’ and ‘breast-cancer-test.csv’ data files we were given, training on the training data and making predictions on the test data using the probability table attained from the training process.

For my implementation the accuracy of the model is at 80%, where it correctly classifies 8/10 instances in the test set (misclassifies instance 1 and 8 in the test set). The rest of this part will be on answering the questions about the data extracted.

1. *“The conditional probabilities $P(X_i = x_i|Y = y)$ for each feature X_i (e.g., age), its possible value x_i (e.g., 10-19), and each class label $Y = y$ (y can be no-recurrence-events or recurrence-events).”*

   I will first look at the conditional probability of each feature for the ‘no-recurrence-events’ class label and then for the ‘recurrence-events’ class label.
   (Data for the probabilities can be found in ‘probabilities.txt’, I’m assuming we just need one xi for each Xi in the report. Have color coded the equivalent xi values for each Y value)

   P(Xi = xi|Y = ‘no-recurrence-events’):
   - P(age = 20-29|class label = ‘no-recurrence-events’) = 0.010 (3dp)
   - P(menopause = ge40|class label = ‘no-recurrence-events’) = 0.458(3dp)
   - P(tumor-size = 30-34|class label = ‘no-recurrence-events’) = 0.159(3dp)
   - P(inv-nodes = 0-2|class label = ‘no-recurrence-events’) = 0.797(3dp)
   - P(node-caps = no|class label = ‘no-recurrence-events’) = 0.874(3dp)
   - P(deg-malig = 2|class label = ‘no-recurrence-events’) = 0.510(3dp)
   - P(breast = left|class label = ‘no-recurrence-events’) = 0.508(3dp)
   - P(breast-quad = left_low|class label = ‘no-recurrence-events’) = 0.366(3dp)
   - P(irradiat = no|class label = ‘no-recurrence-events’) = 0.843(3dp)

   P(Xi = xi|Y = ‘recurrence-events’):
   - P(age = 20-29|class label = ‘recurrence-events’) = 0.011(3dp)
   - P(menopause = ge40|class label = ‘recurrence-events’) = 0.383(3dp)
   - P(tumor-size = 30-34|class label = ‘recurrence-events’) = 0.256(3dp)
   - P(inv-nodes = 0-2|class label = ‘recurrence-events’) = 0.473(3dp)
   - P(node-caps = no|class label = ‘recurrence-events’) = 0.6
   - P(deg-malig = 2|class label = ‘recurrence-events’) = 0.358(3dp)
   - P(breast = left|class label = ‘recurrence-events’) = 0.55
   - P(breast-quad = left_low|class label = ‘recurrence-events’) = 0.386(3dp)
   - P(irradiat = no|class label = ‘recurrence-events’) = 0.613(3dp)

Note:
- *Probabilities.txt contains data for Q1 and 2*
- *Test_ouput.txt contains data for Q3 and the 2nd bulletpoint*

For the probabilities above you can see I have used the same Xi and xi values but different y values to see the difference between no-recurrence and recurrence events. Something interesting about the probabilities above is that the probability of age being 20-29 given that the class label is 'recurrence-events' has almost the same probability as age being 20-29 whilst the class label being 'no-recurrence-events'. This could be due to not having a large enough sample size in the dataset, but if it could be proven statistically significant could be useful for health care workers and patients. This is just one of the ways we can use the Naïve Bayes classifier, and it would be interesting to see how it changes with different datasets.

2. All of this data is visible after running the NaiveBayes.py program on the 'breast-cancer-training' and 'breast-cancer-test' files inside of the probabilities.txt file. This also displays "**2.** *The class probabilities P(Y=y) for each class label Y = y"*. I will briefly go over these values now:

P(Y = 'no-recurrence-events'):

```
no-recurrence-events: 0.7063197026022305
```
[1]

P(Y = 'recurrence-events'):

```
recurrence-events: 0.2936802973977695
```
[2]

Something I should add is that when following the pseudocode from the lectures, we set all values to 1 as default in the calculation. This is called Laplace smoothing/add-one smoothing, and is used to avoid zero probabilities when features haven't been seen in the training data. This is relevant for our models as we have 'xi' with 0 instances. As you can see in the Probabilities above the P(Y = 'no-recurrence-events') = 0.706 and P(Y = 'recurrence-events') = 0.294. This is a good sign as it follows the law of total probability as adding these two together gives a total of 1. The probabilities above are showing that 70% of people in the data were 'no-recurrence-events' while the others were 'recurrence-events'.

3. The last point I will look at is **3.** *"For each test instance, given the input vector **X** = [X_1* = x_1,…, X_9 = x_9], *give the calculated*
   -score(Y = no-recurrence-events,X),
   -score(Y = recurrence-events,X),
   -predicted class of the input vector."

All of the data for this question can be found in the 'test_output.txt' file which is created after running the program. Assuming that our 'breast-cancer-test.csv' files are in the same order, I will count the first index as instance 1 and go up to 10, looking first at the 'no-recurrence-events'.

Original class = no-recurrence-events:

Instance 1:

- score(Y = no-recurrence-events,X) = 4.02e-6
- score(Y = recurrence-events,X) = 7.10e-6
- predicted class of the input vector = recurrence-events

[1]: probabilities.txt, line 1
[2]: probabilities.txt, line 53

Instance 2:
- score(Y = no-recurrence-events,X) = 0.000334
- score(Y = recurrence-events,X) = 2.60e-5
- predicted class of the input vector = no-recurrence-events

Instance 3:
- score(Y = no-recurrence-events,X) = 4.71e-5
- score(Y = recurrence-events,X) = 9.72e-7
- predicted class of the input vector = no-recurrence-events

Instance 4:
- score(Y = no-recurrence-events,X) = 0.000152
- score(Y = recurrence-events,X) = 1.02e-5
- predicted class of the input vector = no-recurrence-events

Instance 5:
- score(Y = no-recurrence-events,X) = 4.35e-6
- score(Y = recurrence-events,X) = 1.92x-6
- predicted class of the input vector = no-recurrence-events

Instance 6:
- score(Y = no-recurrence-events,X) = 0.000600
- score(Y = recurrence-events,X) = 3.88e-5
- predicted class of the input vector = no-recurrence-events

Instance 7:
- score(Y = no-recurrence-events,X) = 0.000207
- score(Y = recurrence-events,X) = 7.42e-5
- predicted class of the input vector = no-recurrence-events

Original class = recurrence-events:

Instance 8:
- score(Y = no-recurrence-events,X) = 0.000315
- score(Y = recurrence-events,X) = 8.93e-6
- predicted class of the input vector = no-recurrence-events

Instance 9:
- score(Y = no-recurrence-events,X) = 3.91e-5
- score(Y = recurrence-events,X) = 6.48e-5
- predicted class of the input vector = recurrence-events

Instance 10:
- score(Y = no-recurrence-events,X) = 4.10e-5
- score(Y = recurrence-events,X) = 5.28e-5
- predicted class of the input vector = recurrence-events

The scores and predicted classes above show where the 80% accuracy of the model comes from. This has to do with the way that the class is predicted in the program, where we calculate the scores for each instance being in each class and which ever classes score is higher is what we classify the instance as. So while the model is working correctly there is still some error, but a classification accuracy of 80% is a good number and means the model likely isn't over fitting or underfitting.

Part 2: Building a Bayesian Network

1. *"Construct a Bayesian Network to represent the above scenario".*

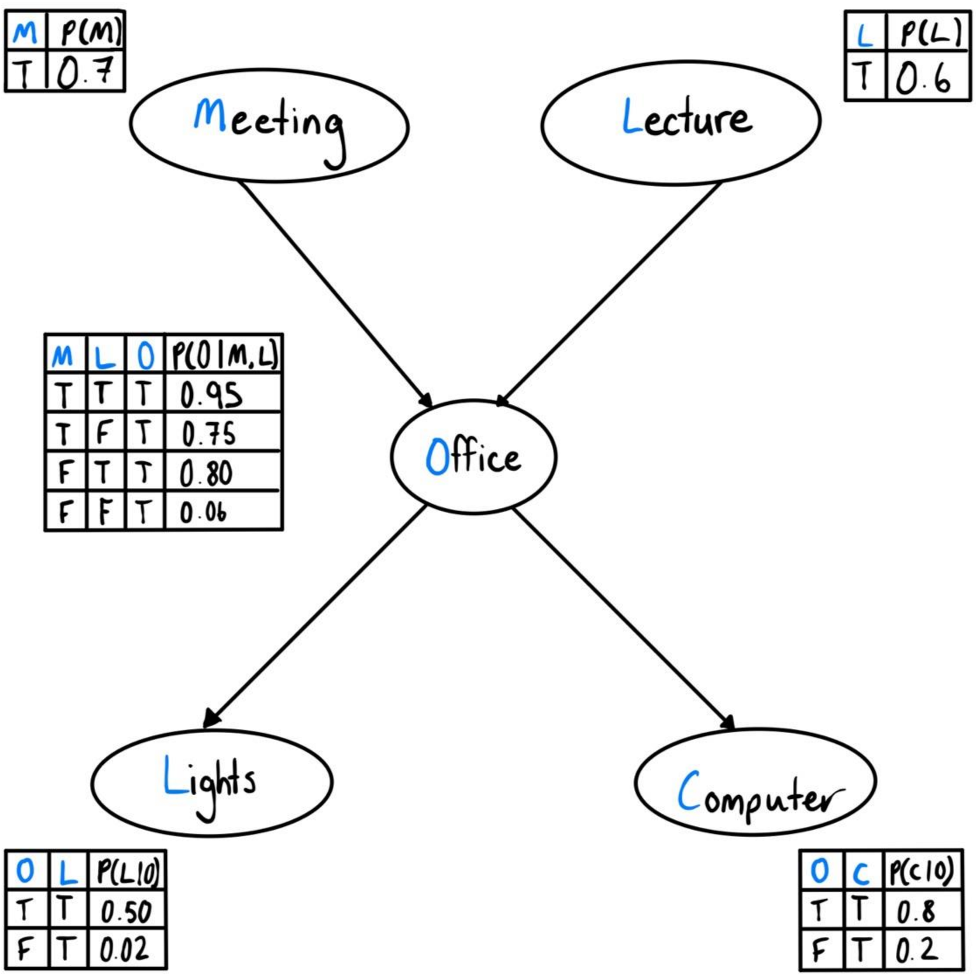
Nodes:

- Meeting – M
- Lecture - L
- Lights – L
- Computer – C
- Office – O

Relationship:

- M/L -> Office = Common effect
- O -> Li/C = Common cause

Bayesian Network:



2. *"Calculate the number of free parameters in your Bayesian network."*

Step 1: Find n states for each node/variable:

- Each variable is binary having 2 states as True or false.

Step 2: Count parameters for each node:

- **Meeting (M): No parents.**

  P(M): |M| -1 = 2 -1 = 1

- **Lecture (L): No parents.**

  P(L): |L| - 1 = 2 – 1 = 1

- **Office (O): Parents = M ^ L.**

  Binary with two parents

  N parent combinations (M,L): 2 x2 = 4.

  Each combination needs a value except for one which can be determined as the remaining probability. This works as probabilities must add to 1.

  Free params = 4 – 1 = 3

- **Light(Li) given Office**

  N parent combinations (O) = 2

  Each combination requires a probability value except for one.

  Free params = 2 – 1 = 1

- **Computer (C) given Office.**

  Same as Light

  Free parameters = 2 – 1 = 1

Sum of Free Parameters:

- 1 + 1 + 3 + 1 + 1 = 7

3. *"What is the probability that Eve has lectures, has no meetings, she is in her office and logged on her computer but with lights off."*

$$\boldsymbol{P(L, \neg M, O, C, \neg Li)}$$

Chain rule:

1. **Change order so that the "causes are always before the effects":**

   $\mathrm{P(L, \neg M, O, C, \neg Li) = P(\neg Li, C, O\ , \neg M, L)}$

2. **Apply Chain Rule:**

   $\mathrm{P(\neg Li, C, O\ , \neg M, L) = P(\neg Li|O)}P(C|O)P(O|\mathrm{\neg M, L})P(\mathrm{\neg M})P(L)$

3. **Substitute values to get final result:**

   $\mathrm{P(\neg Li, C, O\ , \neg M, L)} = 0.5 * 0.8 * 0.8 * 0.3 * 0.6$

   $\mathrm{P(\neg Li, C, O\ , \neg M, L)} = 0.0576$

   $\mathrm{P(\neg Li, C, O\ , \neg M, L)} = 5.76\%$

4. *"Calculate the probability that Eve is in the office."*

Using the **law of total probabilities**, with the following:

Probabilities:

- $\mathrm{P(M)} = 0.7$
- $\mathrm{P(\neg M)} = 1 - 0.7 = 0.3$
- $\mathrm{P(L)} = 0.6$
- $\mathrm{P(\neg L)} = 1 - 0.6 = 0.4$
- $\mathrm{P(O\ |M, L)} = 0.95$
- $\mathrm{P(O\ |}M, \mathrm{\neg L)} = 0.75$
- $\mathrm{P(O|\neg M, L)} = 0.80$
- $\mathrm{P(O|\neg M, \neg L)} = 0.06$

Equation:

- $\mathrm{P(O)} = (P(O|M, L)P(M)P(L) + P(O|M, \mathrm{\neg L)P(M)P(\neg L)} +$
  $\mathrm{P(O|\neg M, L)P(\neg M), P(L) + P(O|\neg M, \neg L)P(\neg M)P(\neg L)}$

Substitution and final result:

- P(O) = (0.95 x 0.7 x 0.6)+(0.75 x 0.7 x 0.4) + (0.80 x 0.3 x 0.6) + (0.06 x 0.3 x 0.4)
- P(O) = 0.399 + 0.21 + 0.144 + 0.0072 = 0.7602

5. *"If we know that Eve is in her office, what is the probability that she is logged on, but her light is off."*

$$\boldsymbol{P(C, \neg Li | O)}$$

Definition of conditional probability:

- $\mathrm{P(C, \neg Li|O)} = \frac{\mathrm{P(C, \neg Li, O)}}{\mathrm{P(O)}}$
- $\mathrm{P(O)}$ = 0.7602 (from Q.4)
- $\mathrm{P(\neg Li|O)}$ = 0.5
- $\mathrm{P(C|O)} = 0.8$

Use Chain Rule to get $\mathbf{P(C, \neg Li, O)}$:

- $\mathrm{P(C, \neg Li, O)} = \mathrm{P(\neg Li|O)P(C|O)P(O)}$
- $\mathrm{P(C, \neg Li, O)} = 0.5$ x 0.8 x 0.7602
- $\mathrm{P(C, \neg Li, O)} = 0.30408$

Substitute back into equation, get final answer:

- $\mathrm{P(C, \neg Li|O)} = \frac{\mathrm{P(C, \neg Li, O)}}{\mathrm{P(O)}}$
- $\mathrm{P(C, \neg Li|O)} = \frac{0.30408}{0.7602}$
- $\mathrm{P(C, \neg Li|O)} = 0.4\ (no\ rounding)$
- $\mathrm{P(C, \neg Li|O)} = 40\%$