COMP307/AIML420 — Fundamentals of AI
Assignment 3: Reasoning under uncertainty
10% of Final Mark

Due date: 11:59 PM - May 17, 2023 (Wednesday)

Objectives

The goal of this assignment is to help you understand the basic concepts and algorithms related to reasoning under uncertainty and probability. These topics are (to be) covered in lectures 14–17. The textbook and online materials can also be checked.

IMPORTANT. You most not use external libraries (like PyTorch, sklearn, or any others) or any AI Tool (like ChatGPT, CoPilot) to complete this assignment.

1 Question Description

In the following, unless explicitly specify, a *capital letter* (e.g., A, B, X, Y) represents a *random variable*, and a *lowercase letter* (e.g., a, b, x, y) represents a value.

Part 1: Reasoning Under Uncertainty Basics [30 marks]

This part contains several questions about the basics of reasoning under uncertainty. You need to write your answers to each of these questions in your report, and **show your working.**

For calculations, you need to show the steps in the form like $P(A=0|B=1) = \frac{P(A=0,B=1)}{P(B=1)}$, to demonstrate that you *know how to calculate* them.

For proving, you also need to clearly show each step of the proof.

Question 1 [15 marks]

The tables below give the prior distribution $P(X)$, and two conditional distributions $P(Y|X)$ and $P(Z|Y)$. It is also known that Z *and* X *are conditionally independent given* Y. All the three variables (X, Y, and Z) are binary variables.

X	$P(X)$
0	0.35
1	0.65

X	Y	$P(Y\|X)$
0	0	0.10
0	1	0.90
1	0	0.60
1	1	0.40

Y	Z	$P(Z\|Y)$
0	0	0.70
0	1	0.30
1	0	0.20
1	1	0.80

1. Compute the table of the joint distribution $P(X, Y, Z)$. **Show the rule(s) you used, and the steps of calculating each joint probability**.

2. Create the full joint probability table of X and Y, i.e., the table containing the following four joint probabilities $P(X=0, Y=0)$, $P(X=0, Y=1)$, $P(X=1, Y=0)$, $P(X=1, Y=1)$. **Show the rule(s) used, and the steps of calculating each joint probability**.

3. From the above joint probability table of X, Y, and Z, calculate the following probabilities. **Show your working**.

(a) $P(Z = 0)$,

(b) $P(X = 0, Z = 0)$,

(c) $P(X = 1, Y = 0|Z = 1)$,

(d) $P(X = 0|Y = 0, Z = 0)$.

Question 2 [15 marks]
Consider three Boolean variables A, B, and C (can take t or f). We have the following probabilities:

- $P(B = t) = 0.7$
- $P(C = t) = 0.4$
- $P(A = t|B = t) = 0.3$
- $P(A = t|C = t) = 0.5$
- $P(B = t|C = t) = 0.2$

We also know that A *and* B *are conditionally independent given* C. Calculate the following probabilities. **Show your working.**

1. $P(B = t, C = t)$
2. $P(A = f|B = t)$
3. $P(A = t, B = t|C = t)$
4. $P(A = t|B = t, C = t)$
5. $P(A = t, B = t, C = t)$

Question 3 [for AIML420 ONLY, 10 marks]
Prove the following statements. **Show your working.**

1. If $P(A|B, C) = P(B|A, C)$, then $P(A|C) = P(B|C)$
2. If $P(A|B, C) = P(A)$, then $P(B, C|A) = P(B, C)$
3. If $P(A, B|C) = P(A|C) * P(B|C)$, then $P(A|B, C) = P(A|C)$

Part 2: Naive Bayes Method [40 marks]

This part is to implement the Naive Bayes algorithm, and evaluate the program on the breast cancer dataset to be described below. The program should build a Naive Bayes classifier from the training dataset and apply it to the test set.

Dataset Description

The *breast cancer* dataset is obtained from the UCI machine learning library (https://archive-beta.ics.uci.edu/ml/datasets/breast+cancer).

The original dataset consists of 286 instances that belong to two classes: *no-recurrence-events* and *recurrence-events*.

Each instance is described by 9 categorical attributes (features). The name and domain of each attribute is described as follows:

1. **age** (9 values): 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99
2. **menopause** (3 values): lt40, ge40, premeno
3. **tumor-size** (12 values): 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59
4. **inv-nodes** (13 values): 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39
5. **node-caps** (2 values): yes, no
6. **deg-malig** (3 values): 1, 2, 3
7. **breast** (2 values): left, right
8. **breast-quad** (5 values): left_up, left_low, right_up, right_low, central
9. **irradiat** (2 values): yes, no

The dataset has some missing values. After removing the instances with missing values, there are 277 instances remaining. Then, these instances are split into the following training and test datasets as follows.:

- **267 training instances**: 189 *no-recurrence-events* + 78 *recurrence-events*.
- **10 test instances**: 7 *no-recurrence-events* + 3 *recurrence-events*.

The datasets are provided in the `breast-cancer-training.csv` and `breast-cancer-test.csv` files.

Requirements

Your job is to use the Naive Bayes classifier to classify the test instances in the `breast-cancer-test.csv` file.

The pseudo code of the training is given as follows to obtain the (conditional) probabilities of each feature given the class, and the probabilities of each class.

Algorithm 1: Training of the Naive Bayes Classifier

```
Input: The training set.
Output: A probability table.
// Initialise the count numbers to 1.
for each class label y do
    count(y) = 1;
    for each feature X_i do
        for each possible value x_i of feature X_i do
            count(X_i, x_i, y) = 1;

// Count the numbers of each class and feature value based on the training
   instances.
for each training instance [X_1 = x_i, ..., X_n = x_n, Y = y] do
    count(y) = count(y) + 1;
    for each feature X_i do
        count(X_i, x_i, y) = count(X_i, x_i, y) + 1;

// Calculate the total/denominators.
class_total = 0;
for each class label y do
    class_total = class_total + count(y);
    for each feature X_i do
        total(X_i, y) = 0;
        for each possible value x_i of feature X_i do
            total(X_i, y) = total(X_i, y) + count(X_i, x_i, y);

// Calculate the probabilities from the counting numbers.
for each class label y do
    prob(y) = count(y)/class_total;
    for each feature X_i do
        for each possible value x_i of feature X_i do
            prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y);

return prob;
```

For the prediction of each test instance, you need to calculate the score of the test instance for each class, and predict the class with the largest score. The score of a class is calculated as follows.

Algorithm 2: Calculation of the class score.

```
Input: A test instance [X_1 = x_1, ..., X_n = x_n], a class label y, the probability table prob.
Output: The score.
score = prob(y);
for each feature X_i do
    score = score * prob(X_i, x_i, y);

return score;
```

You should implement the Naive Bayes method from scratch (not call it from any machine learning library). Your program should take two file names as command line arguments, construct a classifier from the data in the first file, and then apply the classifier to the data in the second file.

You may write the program code in `Java, Python, R, C/C++`, or any other programming language. You should submit the following files electronically and also a report.

- (30 marks) `Program code` for your Naive Bayes Classifier (both the source code and the executable program running on ECS School machines),
- (2 marks) `sampleoutput.txt` containing the output of your program on the test dataset, and
- (8 marks) `A report` in PDF, text or DOC format. The report should include:

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*).
2. The class probabilities $P(Y = y)$ for each class label $Y = y$.
3. For each test instance, given the input vector $\mathbf{X} = [X_1 = x_1, \ldots, X_9 = x_9]$, give the calculated
   - $score(Y = \textit{no-recurrence-events}, \mathbf{X})$,
   - $score(Y = \textit{recurrence-events}, \mathbf{X})$,
   - predicted class of the input vector.

Part 3: Building Bayesian Network [30 marks]

This part is to build a Bayesian Network for the problem described below.

Problem Description

Dr. Rachel Nicholson is a Professor, who lives far away from her university. So, she prefer to work at home and she only comes to her office if she has research meetings with her postgraduate students, or teaching lectures for undergraduate students, or she has both meetings and teaching:

- The probability for Rachel to have meetings is 70%, the probability of Rachel has lectures is 60%.
- If Rachel has both meetings and lectures, the probability of Rachel comes to her office is 95%.
- If Rachel only has meetings (without lectures), the probability of Rachel comes to her office is 75% because she can Skype with her students.
- If Rachel only has lectures (without meetings), the probability of Rachel comes to her office is 80%.
- If Rachel has neither meetings nor lectures, there is a only 6% chance that she comes to the office.
- When Rachel is in her office, half the time her light is off (when she is trying to hide from others to get work done quickly).
- When she is not in her office, she leaves her light on only 2% of the time since the cleaners come for cleaning.
- When Rachel is in her office, 80% of the time she logged onto the computer.
- Because she sometimes work from home, 20% of the time she is not in her office, she is still logged onto the computer.

Note regarding the calculation, you should show your *working process* of the calculation to demonstrate that you *know how to calculate* them.

Requirements

1. Construct a Bayesian network to represent the above scenario. (*Hint: First decide what your domain variables are; these will be your network nodes. Then decide what the causal relationships are between the domain variables and add directed arcs in the network from cause to effect. Finally, you have to add the prior probabilities for nodes without parents, and the conditional probabilities for nodes that have parents.*)

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

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

4. Calculate the probability that Rachel is in the office.

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

Part 4: Bayesian Network: Applications [For AIML420 ONLY, 20 marks]

Identify a real-world application (**different from the examples given in this assignment and the lectures**) that can be described using Bayesian network. There should be at least 5 random variables in this Bayesian network.

In your report, you should:

1. Clearly define the random variables and their domains.
2. Clearly describe their relationships (using plain language).
3. Draw the Bayesian network that can reflect the described relationship.
4. Write the factorisation of the Bayesian network.

2 Relevant Data Files

The relevant data files, information files about the data sets, and some utility program files can be found as a .zip file on the course homepage (See Assignment 2: `https://ecs.wgtn.ac.nz/Courses/COMP307_2023T1/Assignments`).

2.1 Submission Method

The programs and the report should be submitted through the web submission system from the COMP307 or AIML420 course web site **by the due time**. Please make sure you submit to the course you are enrolled in.

Notice that you can submit a .zip file to preserve the subdirectory structure you might have created to organise your submission.

Please check **again** that your programs can be run on the ECS machines easily according to your readme. If the tutors can't run your code, you may **lose marks!** Each tutor has a limited amount of time (¡ 5 minutes) to get your code running, so please don't ask them to use Pycharm, IntelliJ IDEA, Visual Studio, etc to run your code. All these IDEs support exporting runnable code.

2.2 Late Submissions

The assignment must be submitted on time unless you have made a prior arrangement with the course **co-ordinator** or have a valid medical excuse. This year, we are using the ECS extension system for all extension requests. Please make a request there if you think you have a valid reason.

2.3 Plagiarism

Plagiarism in programming (copying someone else's code) is just as serious as written plagiarism, and is treated accordingly. Make sure you explicitly write down where you got code from (and how much of it) if you use any other resources asides from the course material. Using excessive amounts of others' code may result in the loss of marks, but plagiarism should result in zero marks!