DS/CMPSC 442: Artificial Intelligence
Fall 2024
Project-4 Bayesian Networks
(Due: Dec 13th, 12:00 AM)

In this project, every student is required to implement the variable elimination algorithm learned during the class using **Python 3.9** (this is important, make sure that your code compiles with Python 3.9).

Files to Submit: As part of the project submission, we expect you to submit a zip file which contains only two files. You will need to create and submit this file for the assignment by yourselves (See Table 1 below for file names and explanation). ***Make sure that you name this file exactly like this.***

File Name	Explanation
File-1: solution_q1.py	Include all functions that could generate answers for Question 1. Make sure that your solution file could be directly run in the terminal (through the command line "python solution_q1.py")
File-2: solution_q2.py	Include all functions that could generate answers for Question 2. Make sure that your solution file could be directly run in the terminal (through the command line "python solution_q2.py")

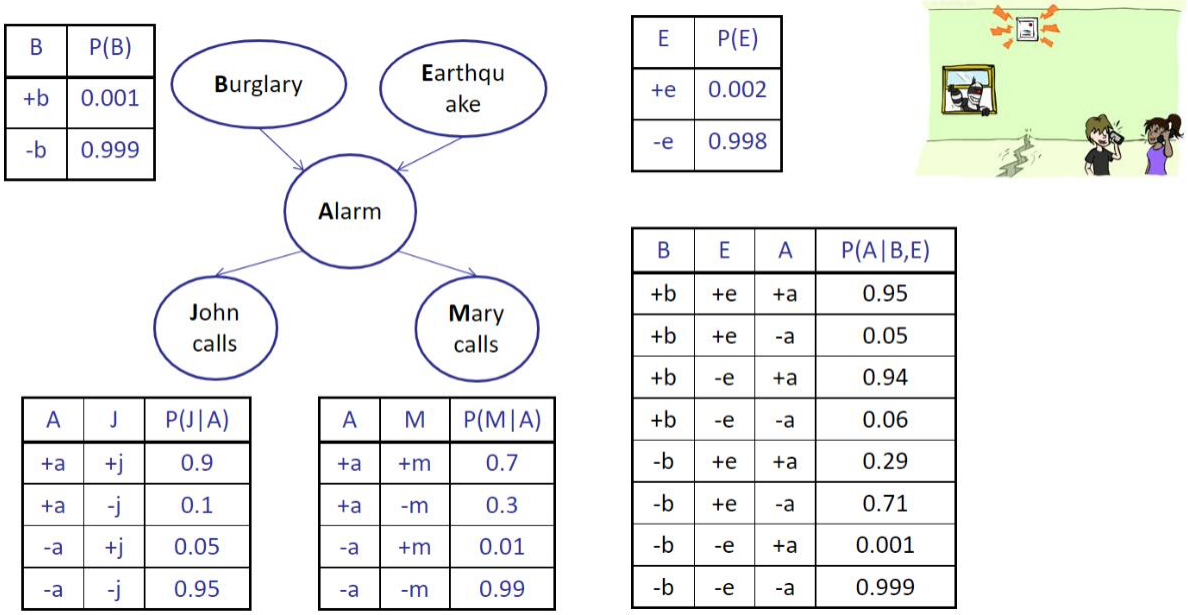
Table 1. Submission files for project-4

Evaluation: Your code will be graded manually by TA for technical correctness. To foster the grading process, they will directly check the output printed in the terminal after running the command "python solution.py" (with different testing version of "input.py" for question 1 that the instructor has created). So, make sure your code can run in the terminal smoothly before making the submission.

Academic Dishonesty: We will be checking your code against other submissions in the class for logical redundancy (using automated software). If you copy someone else's code and submit it with minor changes, we will know. These cheat detectors are quite hard to fool, so please don't try. We trust you all to submit your own work only; please don't let us down. If you do, we will pursue the strongest consequences available to us.

Question-1 Implementing Variable Elimination for Inference in Bayes Nets

You are given the same Bayesian Network which we discussed in class.



B	P(B)
+b	0.001
-b	0.999

E	P(E)
+e	0.002
-e	0.998

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

Assuming this Bayesian network as input, I want you to implement the variable elimination algorithm discussed in class to answer the inference query P(Burglary/John Calls = +j) (meaning what is the probability distribution of the Burglary random variable conditioned on the fact that John actually did call (John Calls = +j).

Of course, it goes without saying but you should be able to easily test whether your variable elimination code is working properly, because you can manually compute P(Burglary/John Calls = +j) using variable elimination (or even, inference by enumeration) in this small sized network.

As a result, you will know very easily whether your code is working correctly or not.

What do we expect to see when we run your code (solution_q1.py)?

Basically, I want you to print the entire probability distribution table P(Burglary/John Calls = +j) (I will leave it up to you to decide how big this table is going to be). As long as you print all the probability values of this probability distribution table, you will get full marks.

Question-2 Implementing a Prediction Model for Diabetes Diagnosis

Project Objective

The goal of this project is to develop a prediction model that determines whether a person has diabetes based on their glucose level (X_1) and blood pressure level (X_2). You will use a supervised dataset containing measurements for 995 individuals, including their glucose levels, blood pressure levels, and a diagnosis of diabetes ($Y = 0$ for no diabetes, $Y = 1$ for diabetes). Use `Part2_Dataset.csv` for this part of the project.

Key Steps Data Splitting

- Split the dataset into **training (70%)** and **testing (30%)** subsets.
- Use a **stratified split** to ensure that the proportions of diabetes ($Y = 1$) and no diabetes ($Y = 0$) in the original dataset are preserved in both subsets.
- Use the training data to learn the probability distribution and Conditional Probability Tables (CPTs) (see Step 2), and reserve the test data for evaluating the model.

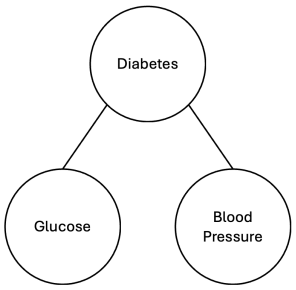
Understanding the Probability Distribution

Follow the steps below to complete the project:

- Assume that the glucose level (X_1), blood pressure level (X_2), and diabetes diagnosis (Y) come from a joint probability distribution $P(X_1, X_2, Y)$.
- Your task is to learn this distribution from the given dataset.
- Once the distribution is learned, you will use it to answer the inference query:

$$P(Y \mid X_1, X_2)$$

- For any new data point (x_1, x_2):
  - Compute $P(Y = 1 \mid x_1, x_2)$ and $P(Y = 0 \mid x_1, x_2)$.
  - Predict **diabetes ($Y = 1$)** if $P(Y = 1 \mid x_1, x_2) > P(Y = 0 \mid x_1, x_2)$; otherwise, predict **no diabetes ($Y = 0$)**.



Question 2.1: Introducing the Conditional Independence Assumption Using the training data, compute the Conditional Probability Tables (CPTs) for:
2.1.1. $P(Y)$: The prior probabilities of diabetes ($Y = 1$) and no diabetes ($Y = 0$).
2.1.2. $P(X_1 \mid Y)$: The conditional probabilities of glucose levels given Y.
2.1.3. $P(X_2 \mid Y)$: The conditional probabilities of blood pressure levels given Y.

Question 2.2: Implementing Inference by Enumeration
2.2.1. Write code to answer the inference query $P(Y \mid X_1, X_2)$.
2.2.2 Generate a lookup table for $P(Y \mid X_1, X_2)$ using the test data.

Question 2.3: Generate Predictions
2.3.1. For each test data point (x_1, x_2):

- Use the table for $P(Y \mid X_1, X_2)$ to compute $P(Y = 1 \mid x_1, x_2)$ and $P(Y = 0 \mid x_1, x_2)$.
- Predict $Y = 1$ if $P(Y = 1 \mid x_1, x_2) > P(Y = 0 \mid x_1, x_2)$; otherwise, predict $Y = 0$.

2.3.2. Compute the model's **accuracy**:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Submit solution_q2.py containing code answering all parts of Question 2. Print the part number (2.1.1, 2.1.2,...) before printing the corresponding answer so as to make the solution easier to read.