Medical Diagnosis

IASD2020/21 Assignment #2

(Version 1.0, December 13, 2020)

Introduction

Medical diagnosis is the process of identifying the underlying cause of a set of symptoms. It is a critical step for determining the right treatment for the patients. It involves not only the patient's symptoms but also tests/exams obtained for identifying specific diseases. This mini-project aims at providing the medical doctors a tool for helping in the patients' diagnosis. It will take into account the symptoms, the possible diseases, and the uncertainty associated with the evidence (tests and exams).

The problem is formulated as follows. It is modeled by an undirected graph, where the nodes are the possible diseases ($\mathcal{D} = \{d_1, \ldots, d_N\}$) and edges are used to represent symptoms. Symptoms can be shared by two or more diseases. The set of edges in the graph is denoted by $\mathcal{C} = \{(d_i, d_j), \ldots\}$ where $d_i, d_j \in \mathcal{D}$. Each edge (d_i, d_j) represent a symptom shared between two different diseases.

Tests/exams are used to assess if a patient has a disease. The set is denoted by $\mathcal{E} = \{e_1, \ldots, e_M\}$, where $M \leq N$, and the map $l : \mathcal{E} \mapsto \mathcal{D}$ specifies the diseases covered by the tests. However, there is uncertainty in the results provided by the tests. Each of these tests/exams is then characterized by two parameters:

True Positive Rate: The probability of testing positive and the patient has the disease;

False Positive Rate: The probability of testing positive and the patient doesn't have the disease.

Consider a set of discrete-time steps, $\mathcal{T} = \{1, \ldots, T\}$, where for each time step, the results of one or more tests/exams are given. Besides, the following propagation law is considered:

- If a patient has disease d_i at time step t, it will continue to have the disease at the instant of time $t+1$;

- If a patient does not have d_i at time step t, it will not have the same disease at the time step $t+1$;

- If a patient has d_i, she/he will have a smaller probability of having other diseases sharing at least one symptom with d_i. We call this the propagation probability.

To conclude, we assume that at the first time instant, $t = 1$, we have absolutely no information (in a probabilistic sense) about which disease the patient has.

The goal of this project is to determine the most probable disease that the patient has, as well as its probability value, at time step T. The decision must consider measurements in the form $\{t, e, z\}$ where $t \in \mathcal{T}$ is the time step, $e \in \mathcal{E}$ is the test result, and $z \in \{True, False\}$ is a boolean representing whether the test came true or positive.

1 Objective

This mini-project aims at solving the previous section's problem, which should be modeled by a Bayesian network, and solved using the variable elimination algorithm for probabilistic inference. The implementation should be done in Python version 3.x. No extra modules, besides the Python Standard Library, are allowed. The search algorithm implementations are the ones from the GitHub repository of the course textbook, namely the module `probability.py` available from `https://github.com/aimacode/aima-python`. A class should be implemented with the name MDProblem, and defines (at least) the following methods (code template):

```python
import probability

class MDProblem:
    def __init__(self, fh):
        # Place here your code to load problem from opened file object fh
        # and use probability.BayesNet() to create the Bayesian network.

    def solve(self):
        # Place here your code to determine the maximum likelihood
        # solution returning the solution disease name and likelihood.
        # Use probability.elimination_ask() to perform probabilistic
        # inference.
        return (disease, likelihood)
```

The code should implement a function called `solve(input_file)` taking as input an opened file object and returning a tuple (disease, likelihood), where the disease is the most likely one, with probability likelihood.

2 Input and Output formats

2.1 Input file

The problem is specified in a text file format where each line contains a list of space-separated fields, where the first field indicates the properties in the line:

`D <d1> <d2> ...`
where `d1`, `d2`, . . . , specify the diseases.

`S <code> <d1> <d2> ...`
is used to specify the symptoms. `code` denotes the symptom, and `d1`, `d2` ... are the diseases associated to the symptom `code`.

`E <code> <d> <TPR> <FPR>`
is used to specify the tests/exams. `code` denotes the test name and `d` the respective disease. `TPR` and `FPR` are the true and false positive rates. Note: one test can only be used to check is a patient has one disease.

`M <e1> <value1> <e2> <value2> ...`
is used to specify the result of a tests/exams, with code `e` (which is the test/exam involved) and `value` with values True or False, identifying whether the test came positive or negative, respectively.

`P <p>`
is used to specify the propagation probability (namely `p`).

Multiple measurement lines may be provided, each one corresponding to a time step, starting at 1 and incrementing for each subsequent line. Therefore, T is given by the total amount of measurement lines.

3 Evaluation

The deliverable for this mini-project has two components:

- A single Python file, called solution.py, implementing the above mentioned MDProblem class, and
- A report in the form of a short questionnaire.

Both components are submitted to the Moodle platform. Instructions are available at the course webpage. The grade is computed as follows:

- 30% from the public tests;

- 30% from the private tests;
- 30% from the questionnaire; and
- 10% from the code structure.

Deadline: Friday, **8-Jan-2021**. No extensions will be possible due to the day of the exam. Projects submitted after the deadline will not be evaluated.

4 Example files

Will be available soon.