Assignment 2: Two probability-based models

The goal of the assignment is to help the student get familiar with two probability-based models. It consists of two tasks: (1) to implement the filtering and prediction algorithm, and (2) to extract the Bayesian Network probability distribution from data.

Task 1: filtering and prediction inference on HMM.

You will work on the **umbrella & weather task**, which have been discussed a lot in the class, and which is also explained clearly in the textbook. It says that a secret guard living in the underground installation wants to know whether it's raining today by observing the director coming in the office with or without an umbrella.

It is a hidden Markov model. Observed evidence variable E represents whether or not the director takes an umbrella. Hidden state variable X represents the weather, which can be raining or sunny.

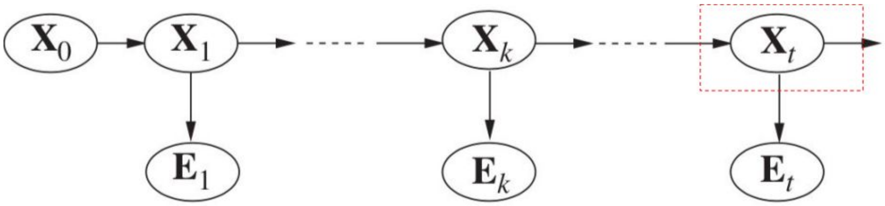
Transition model	today rain	today sunny
yesterday rain	0.7	0.3
yesterday sunny	0.3	0.7

Sensor model	take umbrella	no umbrella
rain	0.9	0.1
sunny	0.2	0.8

Filtering inference on HMM.

In this subtask, you will work on Filtering.py.

You will implement a forward algorithm to do filtering inference, i.e., identify the current state probability distribution given all evidence to date. Please reference slides and section 15.2 in the textbook AMIA.



You are provided with 100 days of the director's behavior data (with/without an umbrella) in **assign2_umbrella.txt**. You need to produce the filtering inference based weather probability distribution from day 1 to day 100.

- Take day 1 for an example, after knowing evidence data of day 1, E_1 , and prior probability distribution at day 0, $P(X_0)$, your algorithm should give weather probability distribution of day 1.
- Take day 65 for another example, after knowing evidence data from day 1 to day 65, and prior probability distribution at day 0 $P(X_0)$, your algorithm should give weather probability distribution of day 65.

Implementation: you need to fill the codes in **filtering()** so that the main function can output correct weather probability distribution.

Prediction inference on HMM.

In this subtask, you will work on Prediction.py.

You will implement a forward algorithm to do prediction inference, i.e., identify a **future** state probability distribution given all evidence to date. Please reference slides and section 15.2 in the textbook AMIA.

You are provided with 100 days of the director's behavior data (with/without an umbrella) in **assign2_umbrella.txt**. You need to produce the prediction inference based weather probability distribution from day 101 to day 150.

- Take day 101 for an example, after knowing evidence data from day 1 to day 100, and prior probability distribution at day 0, $P(X_0)$, your algorithm should give weather probability distribution prediction inference of day 101.
- Take day 145 for another example, after knowing evidence data from day 1 to day 100, and prior probability distribution at day 0 $P(X_0)$, your algorithm should give weather probability distribution of day 145.

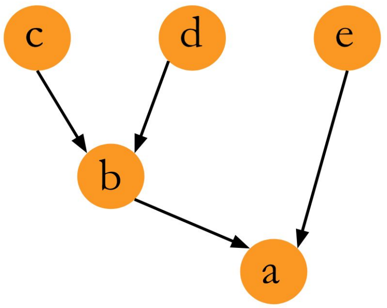
Implementation: you need to fill the codes in **prediction()** so that the main function can output correct weather probability distribution.

Tips: prediction() can call filtering().

Task 2: Extract the Bayesian Network probability distribution from data.

In this task, you will work on BayesianNetwork.py.

For the given Bayesian Network structure, as shown below, please generate the probability distribution of each node from the given data in **assign2_BNdata.txt**.



Joint probability: P(a,b,c,d,e)=P(a|b,e)P(b|c,d)P(c)P(d)P(e)
a has 2 values
b has 3 values
c has 3 values
d has 2 values
e has 2 values

You need to implement **get_p_b_cd() and get_p_a_be()**.

Requirements and Reminders:

1. You **CANNOT** change the classes' names or the required methods' names. However, you can add new variables, constants, and methods in these classes and create new classes if necessary.
2. You are **NOT** allowed to change anything in the **main function** for running your assignment 2 in two tasks.
3. You **CAN ONLY** use Python in this assignment.
4. You **CANNOT** use external Python packages.

Grading:

Your submission will be graded based on:

1. The correctness of the implementation of the required functions (70%)
2. The efficiency of your implementation, make sure your code finishes processing two collections within 2 minutes (20%)
3. Necessary program annotation and commentaries (10%)

Submission Requirements

A zipped file package with the naming convention as "pittids_a2". For example, suppose the Pitt id is jud1, then the submission package should be jud1_a2.zip.
The file package should contain:

1. All the scripts/programs you used for this assignment (**src folder**)
2. Your output in the screen. (This should in **txt file**.)