Assignment 3, Bayes network

Representation of bayesNetwork

For representation of bayesNetwork we created two objects – BayesNetwork and Nodes

Nodes explanation

There are 3 nodes objects all inheriting from class Node
Each node holds 3 fields.

1. Name - string representation
2. Legal values – list of legal assignments to allow to iterate over enumarteAsk algorithms
3. Probability table – showing the probabilistic value for each of its parents nodes legal assignments

Node types and values

1. WeatherNode - legal Values are mild, stormy, Extreme
   Name - WeatherNode
2. BlockedNode – Legal values are True/False
   Name - blockage_VertexName
3. EvacueeNode –Legal values are True/False
   Name – evacuee_VertexName

BayesNetwork explanation

We created the object BayesNetwork inheriting from Singleton to allow a single occurrence of bayesnetwork in the assignment.

The object holds:

1. Graph – a representation of the world (was made for previous assignments)
2. WeatherNode – Node object
3. blockedNodes – a list consisting of all BlockedNodes
4. EvacueeNodes – a list consisting of all EvacueeNodes
5. ChildrenDict – a dictionary with a key for each node, where the value for each key is a list of all its immediate children nodes.
6. ParentsDict – a dictionary with a key for each node, where the value for each key is a list of all its immediate parents nodes.

Functions used –

addRelattion – used to update childrenDict and parentsDict ,
given two values children node and parent node

We used three methods that allow creation of all nodes – the methods are called in order of relation between the nodes.

createWeatherNode – creates a single weather node with probability of each value explicitly written in graph.csv file

createBlockedNodes – create a single Node for each vertex in the world, with probability of each value also explicitly written in graph.csv file in relation to each value of weather
each node calls the method addRelation between itself and weatherNode

createEvacueeNodes – create a single Node for each vertex in the world, with calculated probability using p1 and p2 also written in graph.csv
each node calls the method addRelation between itself and all of its immediate neighboring vertexes BlockedNodes

Enumeration Algorithm

We chose to implement our reasoning algorithm as Eyal suggested using the two methods given in chapter 14.b – Enumerate-ask and Enumerate-all.

The algorithms were implemented as shown in pseudo code with relevant limitationl;

- On enumerate-ask we call enumerate-all for each legalValue of Node object (as explained previously in regards to reason for each node holding all its possible values)
- We made sure that enumerate-all is called with the current order of nodes – where each node is located after its parents in the list

Execution instructions

1. Editing the world and probabilities

The file graph.csv is read by code from the directory of execution (relative path)

Syntax of graph.csv

- #N, X – X is the number of vertexes in our world
- #VX1, FX2 ; X1 is an integer for Vertex name , X2 Float value of probability of the vertex to be blocked in mild weather . P(Blocked) = Min(X2, 1)
  If value is not given, we assume the probability of blockage is 0
- #EX1, X2, X3, WX4; X1 is an integer for Edge number, X2 and X3 are the vertexes of the edge , X4 is the weight (integer) of the edge
- #W, X1,X2,X3; X1- Float Probability of mild weather , X2- Float probability of stormy weather, X3- Float probability of extreme weather
  X1 +X2 +X3 must equal 1 otherwise code terminates on error
- #P1, X ; X -Float value for probability computation
- #P2, X ; X- Float value for probability computation

2. Running the code

Run code from main.py,
Upon execution part 1 will be printed to console (always)
User input prompt will be shown to determine the behavior of code execution.
First you will be given option to add evidence then compute relevant values .