Introduction to Artificial Inteligence

Programming Assignment 3

Programming assignment - Reasoning under uncertainty

Hurricane Evacuation Problem: Predicting the Blockages

Goals

Probabilistic reasoning using Bayes networks, with scenarios similar to the hurricane evacuation problem environment of programming assignment 1.

Uncerain Hurricane Evacuation Problem - Domain Description

As they try to find their best path, in the real world, evacuation forces may be unable to tell in advance where evacuees are and where blocked roads will occur. There may be evidence which can help, but one cannot be sure until the node and/or edge in question is reached! Not knowing the blockage locations makes it hard to plan an optimal path, so reasoning about the unknown is crucial. According to Murphy's laws, evacuees to be saved are more likely to be near blocked edes. Also, blockage has **persistence**, that is blockage yesterday means that blockage today is more likely.

Thus we have a binary variable B(e,t) standing in for "blocked" for each edge e at time t. The blocking events are assumed independent at time 0, with known distributions. Existence of evacuees to be saved at a vertex is a binary variable Pe(v), noisy-or distributed given the blockages at incident edges at time 0, with pi =(1-qi)= 0.6*1/w(e). All noisy-or node have a leakage probability of 0.001, that is, they are true with probability 0.001 when all the causes are inactive. The leakage **may** be ignored for the cases where any of the causes are active.

Blocking probabilities P(B(e,0)=true) at edges for time 0 are provided in the input. For subesquent times (t=1,2, etc.) they are dependent only on the occurence of blockage at the previous time slice, we will have: P(B(v,t+1)=true |B(v,t)=true))=Ppersistence, where Ppersistence is a user-determined constant (usually between 0.7 and 0.99),and P(B(v,t+1)=true |B(v,t)=false))=P(B(v,0)=true). Such a network of variables indexed by time, that aims at modeling the dynamics of the world, is sometimes called a Dynamic Bayes Network (DBN).

All in all, you have 2 types of variables (BN nodes): blockages (one for each edge and time unit) and evacuees (one for each vertex).

In your program, a file specifies the geometry (graph), and parameters such as P(B(e)=true)). Then, you enter some locations where people and blockages are reported either present or absent (and the rest remain unknown). This is the evidence in the problem. Once evidence is instantiated, you need to perform reasoning about the likely locations of blockages, and evacuees (all probabilities below "given the evidence"):

1. What is the probability that each of the vertices contains evacuees?
2. What is the probability that each of the edges is blocked?
3. What is the probability that a certain path (set of edges) is free from blockages? (Note that the distributions of blockages in edges are NOT necessarily independent given the evidence.)
4. What is the path between 2 given vertices that has the highest probability of being free from blockages at time t=1 given the evidence? (bonus)

Input can be as an ASCII file, similar to graph descriptions in previous assignments, for example:

```
#N 4                ; number of vertices n in graph (from 1 to n)

#V1 F 0             ; Vertex 1, no evacuees for sure
#V2 F 0.4           ; Vertex 2, probability of evacuees 0.4
                    ; Either assume evacuees probability 0 by default,
                    ; or make sure to specify this probability for all vertices.

#E1 1 2 W1 B0.2     ; Edge1 between vertices 1 and 2, weight 1, probability of blocking 0.2 at t=0
#E2 2 3 W3          ; Edge2 between vertices 2 and 3, weight 3
#E3 3 4 W3 B0.4     ; Edge3 between vertices 3 and 4, weight 3, probability of blocking 0.4 at t=0
#E4 2 4 W4          ; Edge4 between vertices 2 and 4, weight 4
                    ; Either assume blocking probability 0 by default,
                    ; or make sure to specify this probability for all edges.
#Ppersistence 0.9   ; Set persistence probability to 0.9
```

Requirements

(Part I) Your program should read the data, including the distribution parameters, which are defined as above. The program should construct a Bayes network according to the scenario, for at least 2 time slices: t=0 and t=1. The program should also allow for an output of the Bayes network constructed for the scenario.

For example, part of the output for the above graph, would be:

```
VERTEX 1:
  P(Evacuees) = 0.2
  P(not Evacuees) = 0.8


VERTEX 2:
etc.

EDGE 1, time 0:
  P(Blocakge 1 | not Evacuees 1, not Evacuees 2) = 0.001
  P(Blocakge 1 | Evacuees 1, not Evacuees 2) = 0.6
  P(Blocakge 1 | not Evacuees 1, Evacuees 2) = 0.6
  P(Blocakge 1 | Evacuees 1, Evacuees 2) = 0.84
etc.
```

(Part II) After the network is fully constructed, you should support querying the user for a set of evidence. We do this by reading one piece of evidence at a time (e.g. "Evacuees reported at vertex 2", and then "No blockage reported at edge 1 at time 0" etc.). The online interactive operations your program should support are:

- Reset evidence list to empty.
- Add piece of evidence to evidence list.
- Do probabilistic reasoning (1, 2, 3), or (1,2,3,4), whichever your program supports, and report the results.
- Quit.

Probabilistic reasoning should be done in order to answer the questions on distribution of blockages, etc., and report on the answers, including all the posterior probabilities. You may use any algorithm in the literature that supports solution of BNs, including simple enumerarion, variable elimination, polytree propagation, or sampling.

Deliverables

1. Source code and executable files of programs.
2. Explanation of the method for constructing the BN and your reasoning algorithm.
3. Non-trivial example runs on at least 2 scenarios, including the input and output.

4. Submit makefile and short description on how to run your program. i.e. how what parameters are passed to the program and how other inputs including at least one full example on how to run your program with actual input.

TBA, 2020.