

Introduction to AI

Practical Assignment 3

Uzi Friedman, Lior Siag

January 2021

1 Construction of DBN and reasoning algorithm

We construct the the dynamic base network by first creating all nodes for representing edges at time $t=0$ and vertices. We do this by iterating over the edges in the graph, and for each edge we first add nodes for its vertices and then for the edge. Nodes for vertices contain their probability of having people (given in the input) and nodes for edges contain their CPT. We keep track of the latest time slice that the network was advanced to, and whenever a quarry requires the network to be advanced we call the `advance_time` method. `advance_time` receives an integer denoting the number of time slices we need to advance the network by, and constructs additional nodes for all edges for the added time slices, connected only to the node representing the edge at the previous time slice. The nodes for edges at later times contain the CPT with the persistence value given in the input.

Reasoning on edge/vertex is done by a call to `enumerate_ask`, which operates as shown in class. We first find the relevant nodes, removing barren nodes iteratively, and call `enumerate_all`. It then computes the conditional probability for each node based on the evidence at that point. Reasoning about a path is done in a similar way - we first convert the path to a list of edges, each at the time it would take to arrive at + the starting time (unless we set a flag to `False` in the query, then we check if the entire path is clear at the starting time). Once we have the list of nodes, we call `enumerate_ask` on each, multiplying the results, and adding that the current node is free into evidence after computing it. Overall we get $P(\text{path is clear} | \text{evidence}) = P(e_0 \text{ clear and } e_1 \text{ clear} \dots | \text{evidence}) = P(e_0 \text{ is clear} | \text{evidence}) * P(e_1 \text{ is clear} | e_0 \text{ is clear and evidence}) \dots$

2 How to run the code

The program loads “test.txt” as the graph for the network and then presents the action menu.

Add Evidence: To add evidence about a vertex, we write “v ID X” where ID is the id of the vertex and X is either “t” for true if there are people there

and "f" for false if there are no people there. To add evidence about an edge, we write "e ID T X" where ID is the id of the edge, T is the time, and X is either "t" for blocked or "f" for clear at the specified time.

Query Vertex: Enter the vertex id and get an array back [T,F] where T is the probability that there are people in that vertex and F that there are no people in the vertex.

Query Edge: Enter an edge id and time, and get an array back [T,F] where T is the probability of the edge being blocked at the specified time, F is the probability of it being free at that time.

Query Path: Enter the starting time of the path, and then the path (vertices ids separated with space) and get back the probability of the path being clear for traversal.