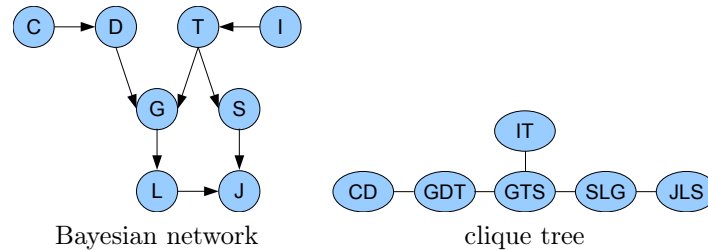


Problem Set 5

Due Sunday February 11, 2018 at 11:55pm

Using the factor code you wrote for the third problem set, write code to perform clique-tree calibration as in algorithm 10.3 (belief propagation). Assume that the clique tree is given (*i.e.* take it and the association of factors to clique-tree nodes as an input to the algorithm).

Use this algorithm to calculate the marginals over each variable in the Bayesian network given below, using the clique tree shown. This network is the same as the running example in the book, except that I (intelligence) has been replaced with T (test-taking ability) and I is now a parent of T, and H (happiness) has been removed.



CDTs:

c^0	c^1		d^0	d^1		t^0	t^1		g^1	g^2	g^3
0.5	0.5	c^0	0.4	0.6	i^0	0.9	0.1	t^0, d^0	0.3	0.4	0.3
		c^1	0.8	0.2	i^1	0.4	0.6	t^0, d^1	0.05	0.25	0.7
								t^1, d^0	0.9	0.08	0.02
								t^1, d^1	0.5	0.3	0.2

i^0	i^1		s^0	s^1		l^0	l^1		j^0	j^1
0.6	0.4	t^0	0.95	0.05	g^1	0.1	0.9	l^0, s^0	0.9	0.1
		t^1	0.2	0.8	g^2	0.4	0.6	l^0, s^1	0.4	0.6
					g^3	0.99	0.01	l^1, s^0	0.3	0.7
								l^1, s^1	0.1	0.9

Your code must be written in C++ and must implement the `calibrate` method of the `bn` class given by `bn.h` and `bn.cpp` (supplied on iLearn with this assignment—these are different than the ones given with assignment 4). The file `caltest.cpp` tests your code's ability to find the marginal over each variable in the network.

This algorithm is pretty straight-forward as the clique tree and mapping from factors to the clique tree are already given. However, make sure your algorithm works in general (and not just for this example as it will be tested on other examples in grading).

This assignment can be completed without changing `bn.h`. **Please do not change `bn.h`**; to do so would change the desired interface and abstraction. You should submit one file: `bn.cpp` Please do NOT submit any additional files.