

Ryland Atkins
Artificial Intelligence
Dr. Allen
Bayesian Network

For this assignment, I implemented an enumeration algorithm using Java. In hindsight, I think it would have been easier to implement the rejection sampling, but I also think I would have learned less about Bayesian Networks had I taken that route. I also prefer the idea of finding an exact probability over finding an estimation of the probability.

The pseudo-code for the meat of my algorithm is as follows.

//Public, not recursive function

P(goals, givens)

 Prob = 1.0

 for goal in goals

 Prob = Prob * *getP(goal, givens)*

 return Prob

//Private recursive function

getP(goal, given)

 if all of goal.parents in given

 return goal.table(parents)

 else

 prob = goal.table(parents)

 for parent in goal.parents

 prob = prob * *getP(parent, given)*

 return prob

The algorithm begins with the Node it wants to find, then traverses the graph upward until it reaches the nodes that were given. One issue with this is that in a cyclic graph you run the risk of getting stuck in an infinite, recursive loop. However, as long as we strictly define our network as an acyclic graph, this will not be a problem.

I was a bit confused about which graph you wanted the four sample outputs from, so I included four outputs from my graph. Since this graph was cyclic, it was a bit more restricting in the possibilities for goal vs. given inputs.

Input	Output
Wat Sbf given Saf Hbt	20.83%
Sat Sbf	63.00%
Wbf Wat Hbt given Hat Sbt	0.95%
Hbf given Sbf Hat Wat Wbt	25.00%