Assignment 2

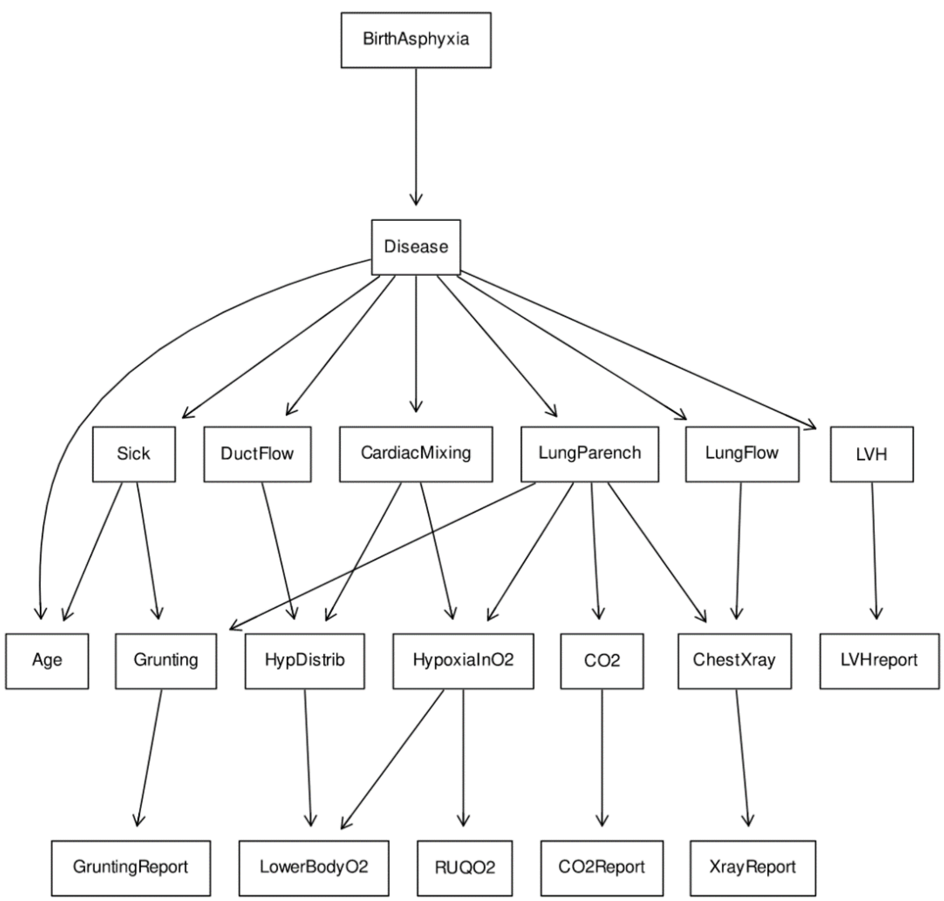
CS 6364 Artificial Intelligence

Deadline – November 20, 11:59 PM

Bayesian Network Representation and Inference

In this assignment, you will use a Bayesian Network (BN) to answer probabilistic queries about a healthcare domain. BNs are probabilistic models that represent the joint distribution over a set of variables. Each BN model consists of two parts – a directed acyclic graph over the variables and local conditional distributions over each variable given its parents.

You are given a BN[1] designed for a hospital that acts as a referral center for newborn babies with congenital heart disease. The BN might be used for preliminary diagnosis to facilitate early appropriate treatment. It contains variables about clinical symptoms, blood gases, electrocardiogram (ECG) and x-ray.



[1] Spiegelhalter, David J., et al. "Bayesian analysis in expert systems." *Statistical science* (1993): 219-247

Starter code

We have provided a set of modules from the AI textbook (Poole and Mackworth) and a code skeleton (hw.py) to get you started. The variable domains and conditional probability distributions for the BN are given as JSON files in the directory named "child." The directory named "earthquake" contains JSON files for a smaller BN called "earthquake." The file "hw_example.py" uses the earthquake BN to demonstrate the representation and inference classes implemented in the provided modules that might be useful for the assignment.

Part 1: Exact Inference

In this part, you will load the BN from the JSON files and perform exact inference on it using the Variable Elimination algorithm.

1. Construct an instance of the BeliefNetwork class. Refer hw_example.py for an example.
2. Implement the perform_exact_inference function. The function computes the query P(Q | E) on a given BN using the Variable elimination algorithm. It eliminates variables in the order specified by the ordering parameter.
3. Use the perform_exact_inference function to compute P(Disease | CO2Report = 1, XrayReport = 0, Age = 0)
    a. Set ordering to be alphabetical order of the variable names
    b. Use a better ordering. You may ask an LLM.
4. Use the timeit module to compute the average time taken over 10 runs by both approaches. Save it as a csv file named "part1.csv" containing exactly one row and two values representing the time taken by each method.

Part 2: Approximate Inference

In this part, you will perform approximate inference using the Rejection Sampling algorithm.

1. Implemented the perform_approximate_inference function. Instead of an ordering, this function takes in the number of samples (n_samples) used to approximate the probability.
2. Use the perform_approximate_inference function to compute the same query as part 1.
    a. Use n_samples = 10
    b. Use n_samples = 100
    c. Use n_samples that would guarantee PAC(epsilon = 0.01, delta = 0.05). Refer https://artint.info/3e/html/ArtInt3e.Ch9.S7.html#SS1.SSSx2

3. Use the timeit module to compute the average time taken by the three approaches. Save it as a csv file named "part21.csv" containing exactly one row and three values representing the time taken by each method.

4. Use the result from part 1 to compute the error in the approximate probability obtained in each of the three approaches. Compute the error as the mean squared error over 10 runs of each method. Save the errors as a csv file named “part22.csv” containing exactly one row and three values representing the error for each method.

Submission

Your submission zip must include:

1. The hw.py script.
2. The csv files part1.csv, part21.csv, part22.csv.
3. A report (pdf) explaining your code and containing your speculations about the results.