1

Bayesian Networks with Exact and Approximate Inference

Braxton McCormack braxton.j.mccormack@gmail.com
Gianforte School of Computing
Montana State University
Bozeman, MT 59715, USA

Ben Heinze ben.c.heinze@gmail.com
Gianforte School of Computing
Montana State University
Bozeman, MT 59715, USA

Abstract

In this project, we will compare the usage and efficiency of two inference methods: an exact inference engine that implements variable elimination, and an approximate inference engine that implements Gibbs Sampling. Both will be used within the context of Bayesian Networks. We will evaluate both inference engines across a set of five Bayesian networks with varying sizes. This project will illustrate the pros and cons of exact inference and approximate inference and determine which algorithm is better in a given scenario.

Keywords: Bayesian Networks, Exact inference, Approximate inference, Variable elimination, Gibbs Sampling

1. System Requirements

The purpose of this project is to use test the difference between an approximate inference algorithm and an exact inference algorithm given five different kinds of Bayesian networks. The following requirements are necessary for this project:

- System must be able to interpret BIF files into Bayesian Networks such that they can be used on the exact-inference algorithm and approximate-inference algorithm. They must not be hard coded in, instead they need a common format.
- System must use an exact-inference algorithm using variable elimination to determine the probability given a specific query.
- System must use an approximate-inference algorithm using Gibbs Sampling algorithm to determine the probability given a specific query.
- System must keep score of how well the exact-inference algorithm and approximate-inference algorithm performs given the same Bayesian network.
- Must test the five given BIF files against both inference algorithms.

We are given five BIF (Bayesian Interchange Format) files to interpret with both inference methods. They all vary in size. The performance of these methods will be evaluated and ranked according. The networks used for testing are listed below.

1. Alarm Network (medium)

- Number of nodes: 37, Number of edges: 46, Number of parameters: 509

- Average degree: 2.49, Maximum number of parents: 4, Average Markov blanket size: 3.51

2. Child Network (small to medium)

- Number of nodes: 20, Number of edges: 25, Number of parameters: 230

- Average degree: 1.25, Maximum number of parents: 2, Average Markov blanket size: 3

3. Hailfinder Network (large)

- Number of nodes: 56, Number of edges: 66, Number of parameters: 2656

- Average degree: 2.36, Maximum number of parents: 4, Average Markov blanket size: 3.54

4. Insurance Network (medium)

- Number of nodes: 27, Number of edges: 52, Number of parameters: 984

- Average degree: 3.85, Maximum number of parents: 3, Average Markov blanket size: 5.19

5. Win95pts Network (large)

- Number of nodes: 76, Number of edges: 112, Number of parameters: 574

- Average degree: 2.95, Maximum number of parents: 7, Average Markov blanket size: 5.92

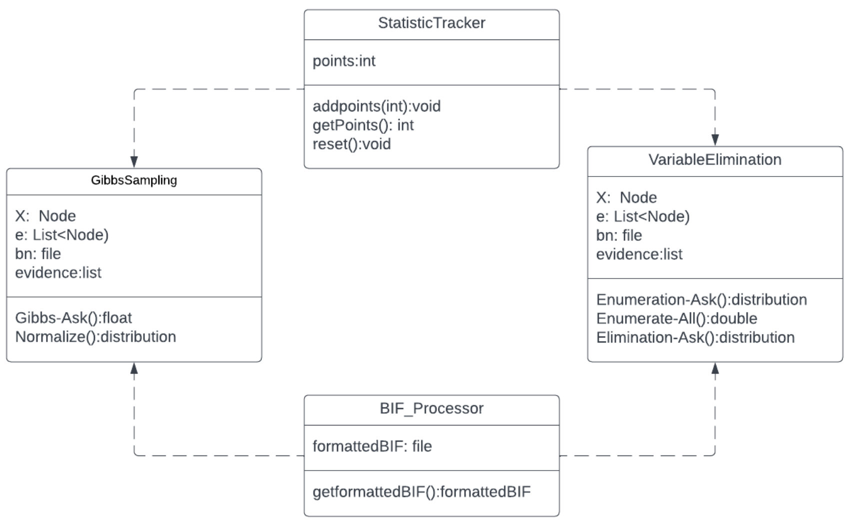
2. System Architecture

VariableElimination The exact inference algorithm that will be interpreting Bayesian Networks. It will return the normalized distribution of a given network.

GibbsSampling The approximate inference algorithm that will be interpreting Bayesian Networks. It will return the normalized distribution of a given network.

BifProcessing Given the five BIF files, we may need to process them such that our two algorithms can properly interpret them. It is not clear whether this is necessary yet.

StatisticTracker This will record the processes made by either inference method.



When planning how we would model this project, we came up with a small-yet-intuitive design. Firstly, since we are not sure whether we need to adjust the BIF files so our networks can interpret it through python, we created a class to fulfil this task on the assumptions that adjustments will be made. If this is not necessary, we will remove the class. Next, we have the StatisticTracker class that we will implement into both our inference algorithms. This allows us to keep score of each algorithm's performance on a given Bayesian network in a simple manner. Once we run every network with and without evidence, we can use each scores to compare one another. Lastly, we have the implementation of the exact inference algorithm and the approximate inference algorithm. For both algorithms, we have a parameter so we can interpret the amount of evidence given by any specific scenario. They both will normalize the derived distributions at the end as well.

3. Test Strategies

It is important to test each of the five given networks in different ways. Therefore, we will test each Bayesian network with different levels of evidence. Alarm, Child, Hailfinder, and Insurance networks will be tested with no evidence, little evidence, and moderate evidence against our two inference algorithms. For the Win95pts Network, we are given a list of problems listed one through six with the following inputs:

- Problem1=No-Output.
- Problem2=Too-Long.
- Problem3=No.
- Problem4=No.
- Problem5=No.
- Problem6=Yes.

4. Task Assignments and Schedule

Format is as follows: Task (initials, internal due date)
bh = Ben Heinze
bm = Braxton McCormack

- Workspace Setup (bh 10/25)
- BIF interpretation(bm, 10/27)
- Variable Elimination (bh, 11/1)
- Gibbs Sampling (bm, 11/1)
- StatisticTracker (bh, 11/2)
- Video demonstration(bm, 11/3)
- Summary Paper(bh,bm 11/4)

References

Russell, Stuart J., et al. "Chapter 13 Probabilistic Reasoning" Artificial Intelligence: A Modern Approach, Fourth Edition, Fourth ed., Pearson Education, Harlow, 2022.