Approximation in Bayesian Networks

Joe Shymanski

October 19, 2023

1 Introduction to the Problem

The problem this paper seeks to tackle can generally be stated as: Given a Bayesian network structure, infer any joint or conditional probability within the network.

Exact inference algorithms are not always feasible. For example, exact inference is almost always impossible in networks with continuous random variables. Additionally, it requires many resources and does not scale up well with the size of the network.

Thus, approximate inference is the next best solution. It is less resource-intensive and scales up much better than its exact counterpart. This paper will investigate three different sampling methods for approximate inference in Bayesian networks: likelihood weighting, Gibbs sampling, and Metropolis-Hastings (MH) sampling.

2 Sampling Method Implementations

2.1 Likelihood Weighting

Likelihood weighting fixes all of the evidence variables. The remaining variables are topologically sorted and then conditionally sampled given their parents. The conditional probabilities of the evidence variables' values are turned into weights and used as counts for that value of the query variable. These counts are then normalized to approximate the probability distribution.

2.2 Gibbs Sampling

Gibbs sampling fixes all of the evidence variables while also randomly assigning values to all the others. It then randomly chooses a non-evidence variable and then conditionally samples a value given its Markov blanket. I used Formula 1 to compute the probability of a node given its Markov blanket, where $Pa(X)$ and $Ch(X)$ return the parents and children of node X, respectively.

$$\frac{P(X=1|Pa(X))\prod_{C\in Ch(X)} P(C|X=1, Pa(C)\setminus\{X\})}{\sum_{x\in X} P(X=x|Pa(X))\prod_{C\in Ch(X)} P(C|X=x, Pa(C)\setminus\{X\})} \tag{1}$$

After each sample, the count of the query variable's value is incremented by one. The resulting counts are normalized to give the approximate probability distribution.

2.3 Metropolis-Hastings Sampling

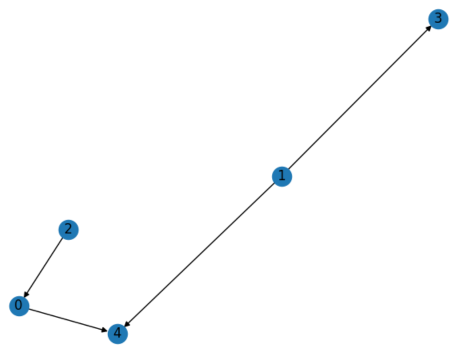
At each iteration, MH sampling performs a Gibbs sample with a certain probability p and a weighted sample otherwise. The higher the p value, the less often a weighted sample is computed. This weighted sample, in essence, represents a resetting of the state. The probability distribution is then approximated using the query variable's value counts in the same way as Gibbs sampling.

3 Results

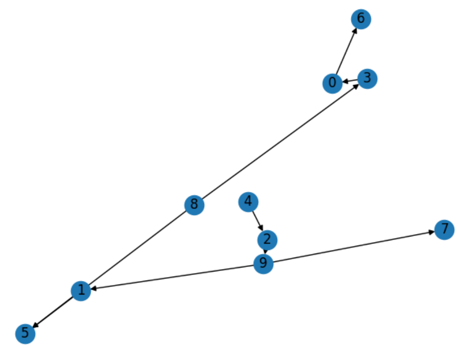
In all comparisons between the approximate inference algorithms, $p = .95$ for MH sampling.

The provided algorithm for exact inference (`enumeration-ask`) was used to calculate the ground truth to compare the approximate results. Since its computation time increases significantly as the network size increases, the networks used in this paper have no more than 10 nodes.

Figure 1 shows the two networks investigated in this paper. Although it is difficult to see, node 5 in Figure 1b has two parents: nodes 1 and 8. The 10-node Bayesian network is investigated more fully, with some parallels drawn in the 5-node network for comparison.



(a) 5 nodes.



(b) 10 nodes.

Figure 1: Bayesian networks.

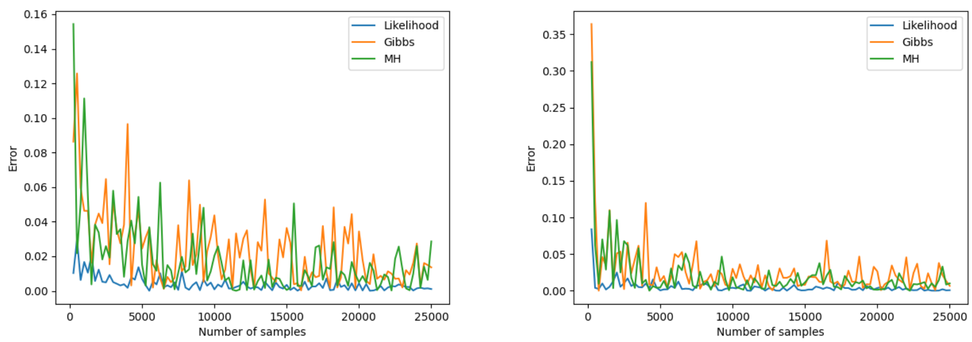
3.1 Upstream Evidence

Figure 2 shows the effects of introducing upstream evidence to the samplers. Its results come from the 10-node network using node 5 as the query variable. The sampling methods in Figure 2b set node 8 equal to `True`. Gibbs and MH appear to have higher variance than likelihood weighting, and they perform much worse with less than 2,500 samples. It does appear that the error from likelihood weighting is usually the lowest. Also, the variances of the Gibbs and MH samplers decrease significantly with upstream evidence.

3.2 Downstream Evidence

According to Figure 3a, there is little difference between the three algorithms on the small network. Here, the query variable is node 1 and no evidence is provided. The trends follow closely with Figure 2a.

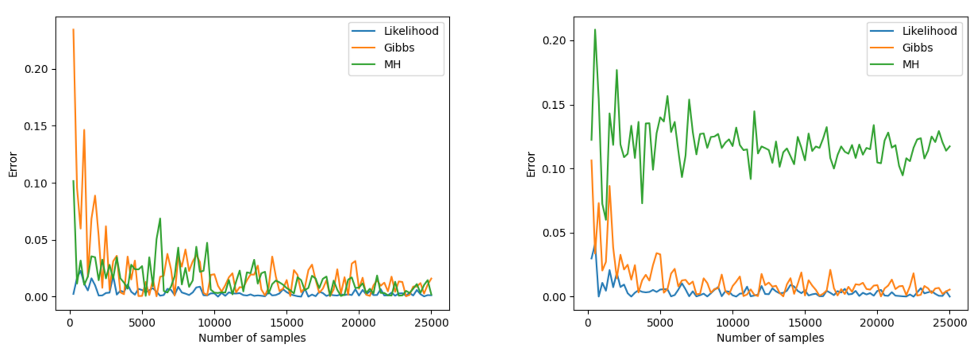
In Figure 3b, there is a significant difference between MH sampling and the other two methods on the small network. The only difference between this and the previous situation is that downstream evidence is provided: nodes 0 and 4 are set to `True`. Gibbs and MH still appear to have higher variance than likelihood weighting.



(a) No evidence.

(b) Upstream evidence.

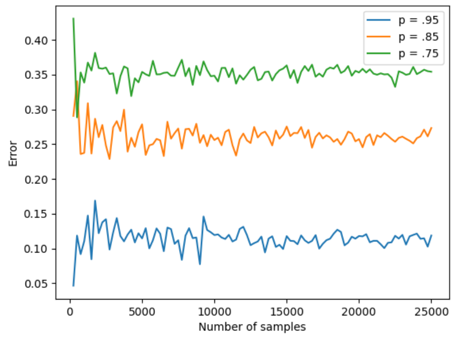
Figure 2: Different methods using query variable 5 on the 10-node network.



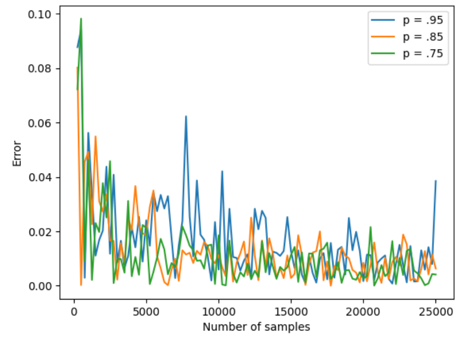
(a) No evidence.

(b) Downstream evidence.

Figure 3: Different methods using query variable 1 on the 5-node network.



(a) Query variable 1 and downstream evidence on the 5-node network.



(b) Query variable 8 and no evidence on the 10-node network.

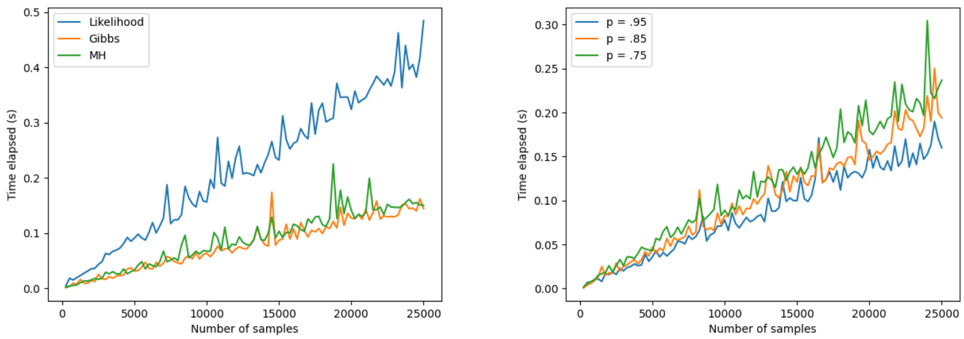
Figure 4: Metropolis-Hastings p comparison.

3.3 Metropolis-Hastings p values

Figure 4a compares p values for the MH sampler in the same scenario as Figure 3b. It shows a clear increase in average error as p decreases. This suggests that restarting more often with a weighted sample is not beneficial for approximation with downstream evidence. In contrast, Figure 4b shows little difference in changing p value for the 10-node network and no evidence.

3.4 Time

The time taken by each algorithm is shown in Figure 5. While these graphs were generated by querying on node 1 with downstream evidence on the 5-node network, their relative trends generalize to all other scenarios, regardless of evidence and network size. Figure 5a clearly shows that likelihood weighting takes significantly more time than Gibbs and MH to complete. Additionally, decreasing the p value in MH sampling appears to slightly increase completion time, according to 5b. All times are positively and linearly correlated with the number of samples.



(a) Different sampling methods.

(b) Metropolis-Hastings p comparison.

Figure 5: Time elapsed using query variable 1 and downstream evidence on the 5-node network.

4 Discussion and Conclusions

My implementation of likelihood weighting seems to achieve the lowest error of any method in every scenario, on average. The Gibbs and MH samplers achieve similar results to each other in most scenarios, except when downstream evidence is present. MH underperforms significantly when presented with downstream evidence, which is unexpected. This could be due to an implementation error in the code. Upstream evidence simply dampens the variance of the Gibbs and MH methods.

Though likelihood weighting performs the best, it consistently takes more time to complete its approximation than the other two methods (around 2-3 times longer). This is expected since likelihood weighting creates an entire weighted sample for every node in the network at every iteration, whereas Gibbs and MH only sample a single variable each time.

The p value in the MH algorithm affects both its completion time and its accuracy. Lower p values mean more weighted samples generated, increasing the total time to complete the approximation. This makes sense due to the reasoning in the above paragraph. Lower p values also increase error with downstream evidence. This could be due to the fact that Gibbs sampling works better with fewer weighted sample "restarts". Gibbs sampling *should* work better with downstream evidence than a weighted sample anyway, even though that was not the case with the implementations in this paper.