Gibbs Sampling vs. Variable Elimination

Group 36
Isaac Walter
Tyler Foster

Abstract

This study implements and compares Variable Elimination and Gibbs Sampling for probabilistic inference in Bayesian Networks (BNs). Evaluating our algos on networks of varying complexity, we demonstrate that VE provides exact solutions efficiently for smaller networks, while Gibbs produces reliable approximations for medium-sized networks but still struggles with larger networks in our case. Our experimental results quantify the trade-offs between computational efficiency and accuracy. These results provide practical guidance for selecting the proper algorithm to select based on specific network characteristics and accuracy requirements.

1 Problem Statement

We currently have three different Bayesian Networks (child, insurance, win95pts) and we need to find the probability of our desired query using variable elimination and Gibbs sampling while simultaneously systematically comparing the 2 algorithms to see which one is the best for our current case. We hypothesize that the variable elimination will compute faster and have a more exact answer compared to Gibbs sampling when we are working with the smaller BN's due to its deterministic behavior; however, Gibbs sampling will work better with larger BN’s. We believe this because variable elimination suffers from exponential time and space complexity as the BN grows in size and connectivity but Gibbs (a Markov Chain method) circumvents this by doing a random walk through the state space. It will provide approximate answers by empirically estimating the probability from the generated samples.

2 Variable Elimination

Variable Elimination will be an "exact inference" and will systematically eliminate variables using factors and summing them out, this format will always guarantee our output to be mathematically correct when evaluating probabilities. The process starts by converting the Bayesian network into factors from each node's probability table. When we have evidence, we first incorporate it by fixing those variables to their observed values which reduces the problem size. Then we choose a good order to eliminate variables using the min-neighbors approach, which helps keep the computation efficient. For each variable we eliminate, we find all factors that contain that variable, multiply them together, and then sum out the variable by adding over all its possible states. We keep doing this until only the query variables remain, then we multiply the final factors and normalize to get our probability distribution.

2.1 Gibbs Sampling

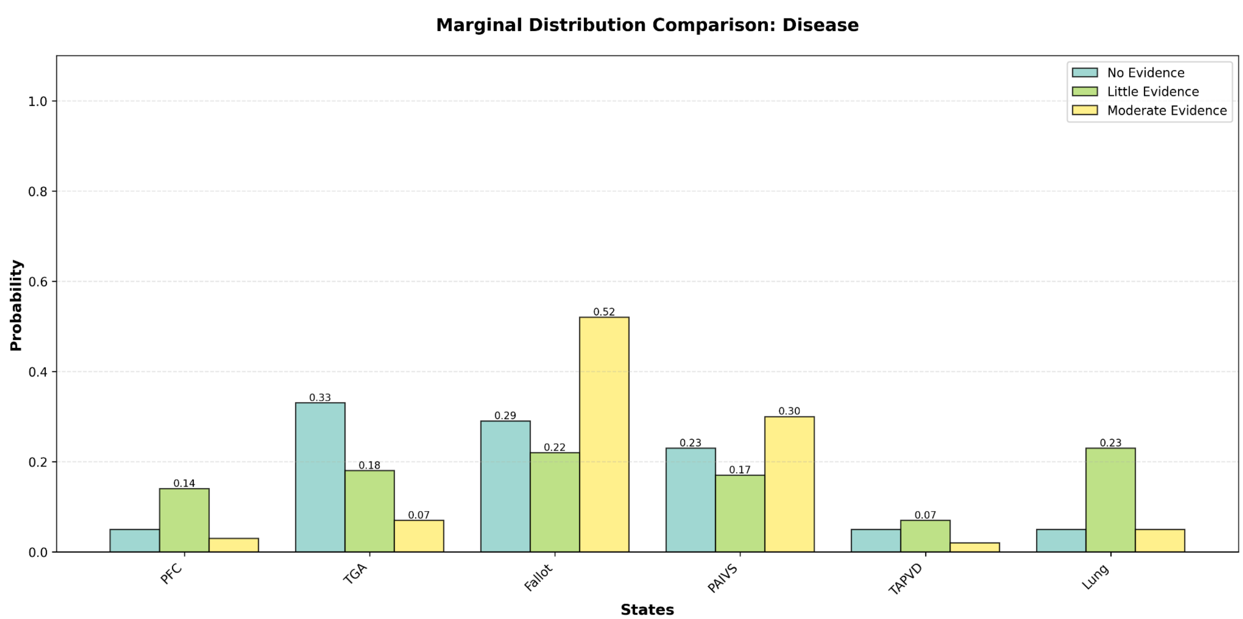
Our Gibbs sampling algorithm is used for performing approximate inference in BN’s. The algo works by iteratively sampling from the conditional distribution of each non-evidence variable given its Markov blanket. It begins by randomly initializing all unobserved variables while fixing the evidence variables to their observed values. Through many sampling iterations, the algo performs a random walk through all the joint probability space, with each variable being updated in turn based on its conditional probability given the current state of all other variables. We use a “burn-in” period to discard a finite amount of samples from the beginning, we assume they are not helpful in this case. The compute_conditional_prob function handles the crucial task of calculating the conditional probability for each variable by multiplying its own conditional probability given its parents with the conditional probabilities of all its children given their respective parents. The final output provides approximate probability distributions for the target query variables, with accuracy improving as more samples are being initiated.

3 Experimental Approach

When creating this project we wanted to be able to compare the two Bayesian inferences algos: exact variable elimination and approximate Gibbs sampling. Our experimental design systematically evaluates their performance across networks of varying complexity to characterize their trade-offs between accuracy and computational efficiency.
We estimate VE as our ground truth for accuracy comparisons, while using timing metrics to assess computational performance (average time per iteration). Testing on our three networks child, insurance, and win95pts with complexity ascending in that order.
Hyperparameters are required for Gibbs sampling and we plan to use trial and error to fine tune them. Since we only have 2 parameters, the total number of samples created and the burn-in size, using the human eye will work fine in this scenario to get the correct tuning due to the simplicity.
Graphs of the most relevant data will be used to visualize our results, to compare them and to provide knowledge on selecting the proper algorithm to select based on specific network characteristics and accuracy requirements.

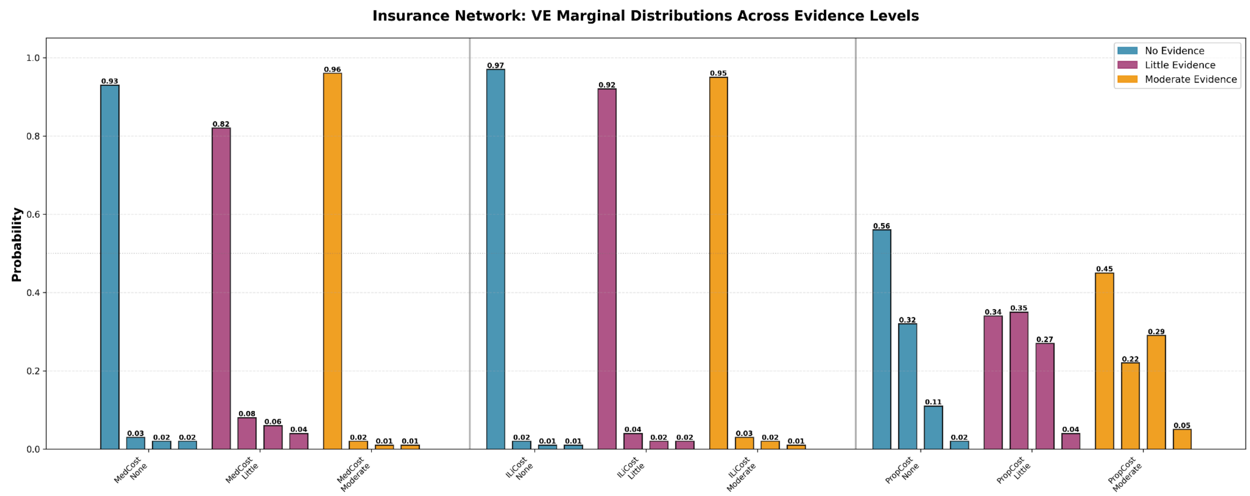
4 Our Results

For Variable Elimination, we ran exact inference across all three networks with varying evidence configurations, producing deterministic marginal probability distributions. These distributions are shown below in Figure 1, Figure 2, and Figure 3



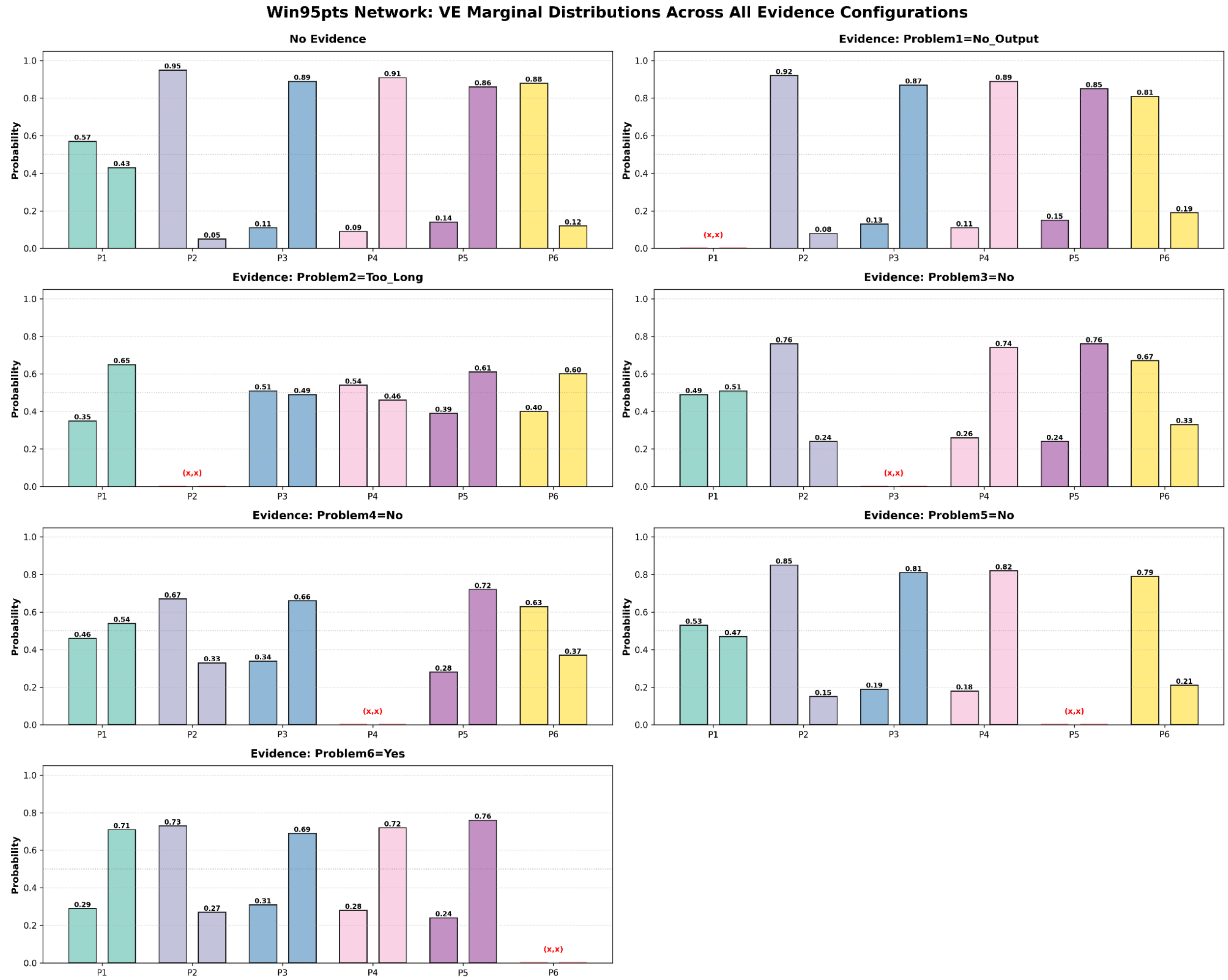
(Figure 1)

Figure 1 shows the child network, which represents a medical diagnostic Bayesian network for rare diseases. Variable elimination on this network provided exact marginal probability distributions for disease variables given specified symptoms in the form of evidence. From Figure 1 we can observe that there is a largely random association between evidence level and probability across the diseases. Testing with little evidence resulted in a smaller range of values for all diseases. Counterintuitively, moderate evidence resulted in a large range, indicating that intermediate levels of information can actually increase the spread of probability. This is likely because moderate evidence is enough to rule out some possibilities while also providing discriminating signals that increase or decrease others.



(Figure 2)

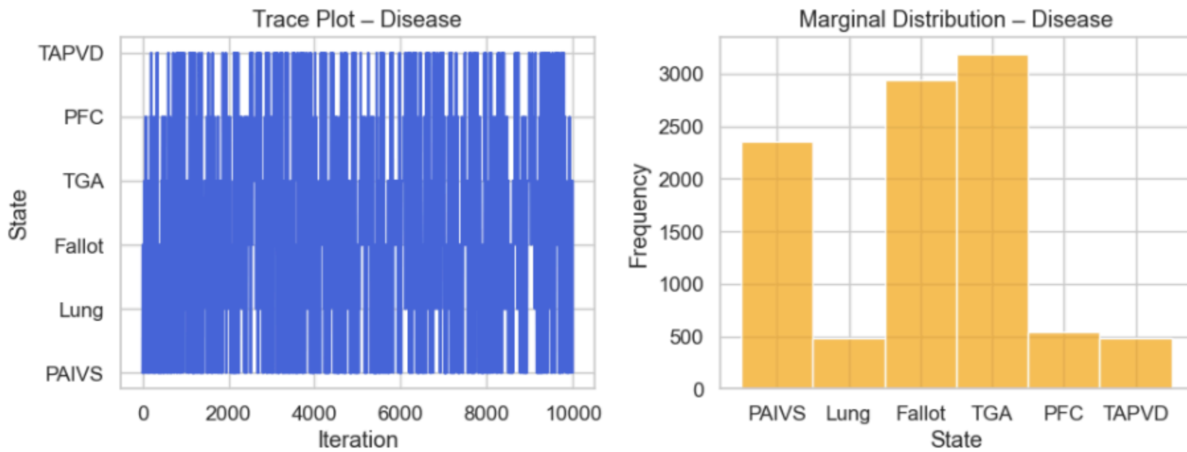
Figure 2 shows that variable elimination on the insurance network with limited evidence produced clear marginal distributions for medical, injury, and property cost. MedCost and ILiCost were both really skewed toward the lower cost tiers, showing that most medical and liability expenses remain low despite an increase in evidence level. Property cost shows a more balanced spread with higher probability of larger costs associated with more evidence. These results show how evidence levels can shape marginal distributions.



(Figure 3)

Figure 3 shows the win95pts network marginal distributions across win95pts related problems. We can see from the graphs how various types of evidence impact the probabilities of different problems. When evidence = {none, Problem1=No_Output, Problem3=No, and Problem5=No}, we observe a relatively large probability of either yes or no for problems 2 through 6. At the same time, problem 1 shows little change amongst these evidence levels. There is a less aggressive probability of either yes or no when evidence: Problem6=Yes. Lastly, the graphs show that when given evidence: {Problem4=No and Problem2=Too_Long}, we see probabilities of either yes or no getting much closer to 50% for all problems (except the evidence problem).

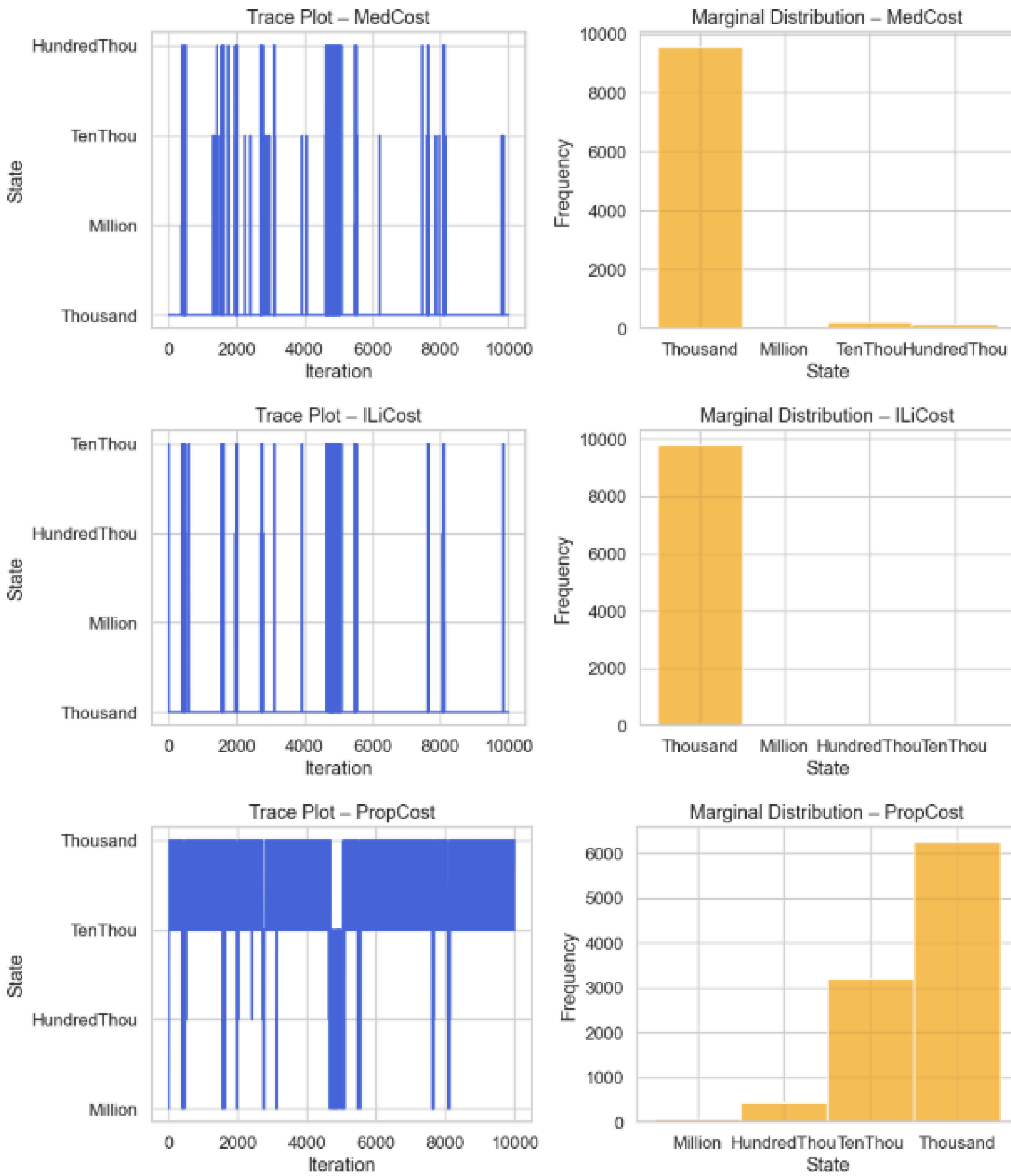
For Gibbs sampling we have created 2 graphs to represent the marginal distribution and a trace plot for convergence (all BN's were run with no evidence).




```
Average time per iteration: 0.00049 seconds
```

(Figure 4)

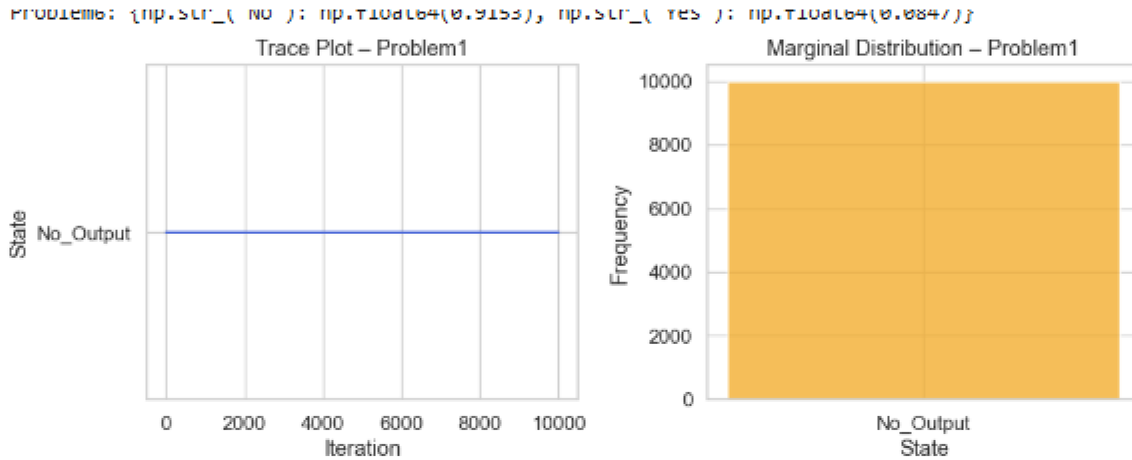
In figure 4 it shows our Gibbs algo working with the child.bif BN. The trace plot shows the changes over the course of the sampling iterations and the marginal distribution shows the total distribution between every state. Another point of data that this provides is the average time per iteration (all tests for gibbs were run on the same hardware to get accurate measures). The trace plot jumps between different states a lot, meaning the Gibbs sampler is exploring the posterior distribution well and not getting stuck on a state. There seems to be stable convergence because the trace plot shows no drifts or trends that would mean otherwise. The density in the trace plot shows what states are going to have the highest distribution values. The marginal distribution aligns with the values given in the project doc meaning it was a full success. Also an iteration time of 0.49 milliseconds is very efficient which we are happy to see.




```
Average time per iteration: 0.00079 seconds
```

(Figure 5)

Figure 5 shows the insurance BN which has 3 query variables and that is why there are 6 graphs. In all 6 graphs there is relatively normal distribution and trace plots like in figure 1 which is a great sign for our algo. Something we expected is the average time per iteration went up and this probably has to do with the larger search space being that the BN is larger which was true.



(Figure 6)

For the largest BN we tested (Win95pts) our Gibbs sampling was clearly getting stuck trying to find the correct distribution as seen in figure 6. We had 6 query variables for this problem but we decided to only show the graphs for one being that they all looked like the one above. Lets analyze the trace plot, it is a straight line through all 10,000 iterations meaning that there were no points in the sample iterations where we could use them as data. This shows that our chain did not explore any search space and there is a definite problem with either stuck variables, deterministic CPT's, or non-convergence.

5 Discussion

Our initial hypothesis was that Variable Elimination would work extremely well on small networks, while the Gibbs sampling algorithm would be ideal for the largest networks. The testing on the first 'child' and 'insurance' network seemed to support this, the marginal distribution was basically a perfect match when compared to the baseline from Variable Elimination. This confirms our logic is at least correct and working as expected. When looking at the evolving iterations of Gibbs (such as in figure 1) we can verify that it is running correctly by frequently transitioning without getting stuck. We expected that the Gibbs algorithm would perform exceptionally on the largest networks but looking at the iterations that was clearly not the case. Instead of switching between states and mixing well it gets stuck instantly due to the network only containing deterministic probabilities - it will have no ability to explore. Variable elimination will have to deal with some extremely large intermediate factors when trying to simplify a network with a wide markov blanket and will get computationally expensive to run. Neither algorithm is perfect and large or complex networks need to be heavily optimized to get good results.

Our Gibbs sampling algorithm proved extremely successful on the small and medium networks, achieving fast performance and having only a small deviation from the "perfect" Variable Elimination. Although we proved it is not foolproof to throw at any large network, it can be tuned to be efficient at medium to large sized networks as long as they are laid out in a fashion that the algorithm can explore (reasonable amount of parents and children as well as enough of a probability spread for the random jumping to explore).

Variable Elimination was predictable and exact - which was absolutely necessary to be able to compare our results to Gibbs sampling without it depending on the luck of the run. As long as the factor size did not get out of control the algorithm was still quick and reliable up to the medium insurance network.

Evidence was beneficial by reducing the search space and shrinking the factors early on for Variable Elimination, but can slow Gibbs sampling if it makes many local conditions nearly deterministic.

6 References

None.