# Computing Probabilities From Bayesian Networks Using Exact and Approximate Inference Algorithms

**Anthony (AJ) Mann** ANTHONYSTEVENMANN@GMAIL.COM
*Artificial Intelligence Student*
*Montana State University*
*Bozeman, MT 59718, USA*

**Jackson Thorn** JTHORN2448@GMAIL.COM
*Artificial Intelligence Student*
*Montana State University*
*Bozeman, MT 59718, USA*

## Abstract

This paper aims to explore the use of exact and approximate inference algorithms to compute conditional probabilities in Bayesian networks. Variable elimination computes exact probabilities in a Bayesian network by successively summing out hidden variables, while Gibbs sampling approximates probabilities by iteratively sampling each variable given the current states of other variables. The accuracy of both algorithms is then compared across three Bayesian networks with varying levels of evidence. We found that both algorithms achieved nearly identical levels of performance in each attempted Bayesian network.

## 1 Introduction

### 1.1 Bayesian Networks

Bayesian networks provide a compact representation of joint probability distributions over sets of random variables. Each node represents a variable, and its edges represent conditional dependencies between variables. Provided with a Bayesian network structure and its conditional probability tables, the fundamental task is to compute posterior probabilities of query variables given the observed evidence. Inference becomes computationally challenging as networks grow in size and complexity.

### 1.2 Algorithms

There are two primary approaches for performing inference in Bayesian networks: exact and approximate methods. Variable elimination systematically computes probabilities by eliminating variables through a sequence of factor operations. Correctness is guaranteed, but may become intractable for larger networks due to exponential growth in intermediate factor sizes. Gibbs sampling estimates probabilities through Monte Carlo sampling methods. Gibbs sampling can handle larger networks and provide accurate estimates as more samples are collected.

### 1.3 Hypothesis

This paper implements and evaluates both variable elimination and Gibbs sampling across three Bayesian networks of varying complexity: the Child network (20 notes), the Insurance network (27 nodes), and the Win95pts network (76 nodes). The Child and Insurance networks are tested under conditions with no evidence, little evidence and moderate evidence, while the Win95pts network is tested across seven different cases. We hypothesize that variable elimination will produce exact results but may struggle with computational efficiency on the larger networks, while Gibbs sampling will provide approximate results that converge toward exact values as iterations increase.

## 2 Descriptions of Implemented Algorithms

### 2.1 Variable Elimination

Variable elimination is an exact inference algorithm that computes posterior probabilities by removing variables from a Bayesian network through factor operations. Factors are defined as multi-dimensional tables that represent probability distributions over sets of variables. Each conditional probability table in the network is initially converted into a factor. The algorithm then eliminates variables one by one until only the query variables remain.

The elimination process consists of four primary operations: factor reduction, factor multiplication, marginalization, and normalization, to ensure the probabilities sum to 1. Factor reduction applies evidence by restricting factors to only those entries consistent with observed variable values. This reduces the size of factors by removing inconsistent assignments. Factor multiplication combines multiple factors that share common variables by multiplying their probability values for each joint assignment. Marginalization sums out a variable from a factor by adding probabilities across all possible values for that variable.

The algorithm uses a greedy min-degree heuristic to determine the elimination order. At each step, the variable appearing in the fewest remaining factors is selected for elimination. As factors are combined during elimination, the heuristic is dynamic and keeps the intermediate factor sizes from blowing up. For the moderately sized networks on which we tested our implementation, this approach worked well in balancing computational complexity and output quality.

### 2.2 Gibbs Sampling

Unlike variable elimination, Gibbs sampling is an approximate inference algorithm that estimates posterior probabilities through Markov Chain Monte Carlo sampling. Sample probabilities are generated from the joint probability distribution by resampling each non-evidence variable conditioned on the current states of all other variables. As the number of these samples increases, its distribution converges to the true posterior probability distribution.

For each variable, we loop through each of its states and calculate a probability conditioned on its Markov blanket. During the calculation of this probability, smoothing takes place. This smoothing helps mellow extremely deterministic probabilities. Once this probability has been computed, it must be normalized such that the probabilities for that variable sum to 1. From here, the algorithm chosen a state for the variable based on the probabilities that were sampled. This new state overrides the initial random state.

Samples for the chosen variable are collected if and only after a burn-in period has elapsed. This burn-in period helps remove potential initial biases in the sampling process that may be influenced by the random initialization. The sampling process will continue for the specified number of iterations and burn-ins. After all samples have been collected, they are used to compute an average which form the final approximate posterior distribution. The number of iterations and burn-in samples are scaled by network size: the larger networks require more samples to converge.

## 3 Experimental Approach

The primary objective is to compare the accuracy of variable elimination and Gibbs sampling across three Bayesian networks of varying complexity. Variable elimination computes exact posterior probabilities by systematically summing out hidden variables according to the network structure, providing deterministic results. Gibbs sampling approximates probabilities by iteratively sampling each variable conditioned on the current state of the others. By comparing the outputs of both algorithms against known values, the reliability and convergence behavior of approximate inference can be assessed against to exact baseline.

Three networks are used for evaluation: the Child network (20 nodes, 25 edges), the Insurance network (27 nodes, 52 edges), and the Win95pts network (76 nodes, 112 edges). Each network is tested under multiple evidence conditions. The Child and Insurance networks are evaluated with no evidence, little evidence (4 observed variables), and moderate evidence (7 observed variables). The Win95pts network is evaluated with no evidence and six distinct single-variable evidence scenarios. For each test case, specific query variables are designated for marginal probability computation. The Child network queries the Disease variable, the Insurance network queries MedCost, ILiCost, and PropCost, and the Win95pts network queries all six Problem variables.

Performance is measured by comparing the marginal probability distributions produced by each algorithm against expected values. For variable elimination, exact matches with ground truth validates correct implementation of factor operations and elimination ordering. For Gibbs sampling, the deviation from exact values gives an approximation error and indicates convergence quality. The number of iterations and burn-in samples for Gibbs sampling are scaled based on network size: 75,000 iterations with 7,500 burn-in for networks under 25 nodes, 100,000 iterations with 10,000 burn-in for networks between 25 and 50 nodes, and 150,000 iterations with 15,000 burn-in for networks with more than 50 nodes.

### 3.1 Implementation

The Bayesian networks are provided in the BIF format, parsed using pgmpy's BIFReader class, which extracts the network structure, variable states, parent relationships and conditional probability tables. This data is converted into a representation such that both variable elimination and Gibbs sampling can access it. Query variables and evidence assignments are determined programmatically based on network name and evidence level parameters.

#### 3.1.1 Variable Elimination

Variable elimination converts each conditional probability table into a factor representation stored as a dictionary mapping variable assignments to probabilities. Factor multiplication iterates over all possible joint assignments of variables in the combined factor, retrieving probabilities from each input factor and computing their product. Marginalization sums probabilities across all values of the eliminated variable, grouping entries by their assignments relative to remaining variables. Evidence reduction filters factor entries to retain only those consistent with their observed values. The greedy min-degree heuristic determines elimination order by repeatedly selecting the variable appearing in the fewest remaining factors, simulating factor combination to track changing variable connectivity throughout the elimination process.

#### 3.1.2 Gibbs Sampling

Gibbs sampling initializes all non-evidence variables to random states and computes the Markov blanket for each variable by identifying its parents, children, and children's parents. During each iteration, variables are processed in a random order. For each variable, conditional probabilities are computed by querying the variable's conditional probability table given current parent states, then multiplying by the conditional probabilities of each child given current parent states. We used a smoothing parameter of 0.001 which bounds all probabilities away from zero and one to prevent deterministic relationships from causing sample problems. Probabilities are normalized and used as weights for random state selection. Samples are collected only for query variables and only after the burn-in period completes. Final probabilities are computed by averaging all collected samples for each variable state.

## 4 Results

Table 1: Comparison of Actual Network Probabilities with Algorithm Results

(a) Child Network — *Disease*

| | *PFC* | | | *TGA* | | | *Fallot* | | | *PAIVS* | | | *TAPVD* | | | *Lung* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.05 | 0.05 | 0.05 | 0.33 | 0.33 | 0.33 | 0.29 | 0.29 | 0.28 | 0.23 | 0.23 | 0.23 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| *Little* | 0.14 | 0.14 | 0.14 | 0.18 | 0.18 | 0.18 | 0.22 | 0.22 | 0.22 | 0.17 | 0.17 | 0.17 | 0.07 | 0.07 | 0.07 | 0.23 | 0.23 | 0.23 |
| *Moderate* | 0.03 | 0.03 | 0.03 | 0.07 | 0.07 | 0.08 | 0.52 | 0.52 | 0.52 | 0.30 | 0.30 | 0.31 | 0.02 | 0.02 | 0.02 | 0.05 | 0.05 | 0.05 |

(b) Insurance Network — *MedCost*

| | *Thousand* | | | *TenThou* | | | *HundredThou* | | | *Million* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.93 | 0.93 | 0.93 | 0.03 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| *Little* | 0.82 | 0.82 | 0.82 | 0.08 | 0.08 | 0.08 | 0.06 | 0.06 | 0.06 | 0.04 | 0.04 | 0.04 |
| *Moderate* | 0.96 | 0.96 | 0.96 | 0.02 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |

(c) Insurance Network — *ILiCost*

| | *Thousand* | | | *TenThou* | | | *HundredThou* | | | *Million* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.97 | 0.97 | 0.97 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| *Little* | 0.92 | 0.92 | 0.92 | 0.04 | 0.04 | 0.04 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| *Moderate* | 0.95 | 0.95 | 0.95 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 |

(d) Insurance Network — *PropCost*

| | *Thousand* | | | *TenThou* | | | *HundredThou* | | | *Million* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.56 | 0.56 | 0.58 | 0.32 | 0.32 | 0.31 | 0.11 | 0.11 | 0.10 | 0.02 | 0.02 | 0.02 |
| *Little* | 0.34 | 0.34 | 0.35 | 0.35 | 0.35 | 0.35 | 0.27 | 0.27 | 0.25 | 0.04 | 0.04 | 0.04 |
| *Moderate* | 0.45 | 0.45 | 0.46 | 0.22 | 0.22 | 0.22 | 0.29 | 0.29 | 0.26 | 0.05 | 0.05 | 0.05 |

Table 2: Win95pts Network — Problem 1 & 2

| | *Problem 1* | | | | | | *Problem 2* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Normal_Output | | | OK | | | No_Output | | | Too_Long | | |
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.57 | 0.57 | 0.58 | 0.43 | 0.43 | 0.42 | 0.95 | 0.95 | 0.94 | 0.05 | 0.05 | 0.06 |
| *P1* | x | x | x | x | x | x | 0.92 | 0.92 | 0.95 | 0.08 | 0.08 | 0.05 |
| *P2* | 0.35 | 0.35 | 0.42 | 0.65 | 0.65 | 0.58 | x | x | x | x | x | x |
| *P3* | 0.49 | 0.49 | 0.63 | 0.51 | 0.51 | 0.37 | 0.51 | 0.51 | 0.88 | 0.49 | 0.49 | 0.12 |
| *P4* | 0.46 | 0.46 | 0.55 | 0.54 | 0.54 | 0.45 | 0.67 | 0.67 | 0.83 | 0.33 | 0.33 | 0.17 |
| *P5* | 0.53 | 0.53 | 0.55 | 0.47 | 0.47 | 0.45 | 0.85 | 0.85 | 0.85 | 0.15 | 0.15 | 0.15 |
| *P6* | 0.29 | 0.29 | 0.31 | 0.71 | 0.71 | 0.69 | 0.73 | 0.73 | 0.65 | 0.27 | 0.27 | 0.35 |

Table 3: Win95pts Network — Problem 3 & 4

| | *Problem 3* | | | | | | *Problem 4* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | No | | | Yes | | | No | | | Yes | |
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.11 | 0.11 | 0.11 | 0.89 | 0.89 | 0.89 | 0.09 | 0.09 | 0.08 | 0.91 | 0.91 | 0.92 |
| *P1* | 0.13 | 0.13 | 0.10 | 0.87 | 0.87 | 0.90 | 0.11 | 0.11 | 0.08 | 0.89 | 0.89 | 0.92 |
| *P2* | 0.51 | 0.51 | 0.45 | 0.49 | 0.49 | 0.54 | 0.54 | 0.54 | 0.52 | 0.46 | 0.46 | 0.48 |
| *P3* | x | x | x | x | x | x | 0.26 | 0.26 | 0.13 | 0.74 | 0.74 | 0.87 |
| *P4* | 0.34 | 0.34 | 0.18 | 0.66 | 0.66 | 0.82 | x | x | x | x | x | x |
| *P5* | 0.18 | 0.18 | 0.21 | 0.82 | 0.82 | 0.79 | 0.28 | 0.28 | 0.19 | 0.72 | 0.72 | 0.81 |
| *P6* | 0.67 | 0.67 | 0.43 | 0.33 | 0.33 | 0.57 | 0.63 | 0.63 | 0.35 | 0.37 | 0.37 | 0.65 |

Table 4: Win95pts Network — Problem 5 & 6

| | *Problem 5* | | | | | | *Problem 6* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | No | | | Yes | | | No | | | Yes | |
| Evidence | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** | **Actual** | **VE** | **Gibbs** |
| *None* | 0.14 | 0.14 | 0.13 | 0.86 | 0.86 | 0.87 | 0.88 | 0.88 | 0.89 | 0.12 | 0.12 | 0.11 |
| *P1* | 0.15 | 0.15 | 0.12 | 0.85 | 0.85 | 0.88 | 0.81 | 0.81 | 0.83 | 0.19 | 0.19 | 0.17 |
| *P2* | 0.39 | 0.39 | 0.34 | 0.61 | 0.61 | 0.66 | 0.40 | 0.40 | 0.46 | 0.60 | 0.60 | 0.54 |
| *P3* | 0.24 | 0.24 | 0.17 | 0.76 | 0.76 | 0.83 | 0.67 | 0.67 | 0.86 | 0.33 | 0.33 | 0.14 |
| *P4* | 0.28 | 0.28 | 0.21 | 0.72 | 0.72 | 0.79 | 0.63 | 0.63 | 0.79 | 0.37 | 0.37 | 0.21 |
| *P5* | x | x | x | x | x | x | 0.79 | 0.79 | 0.81 | 0.21 | 0.21 | 0.19 |
| *P6* | 0.24 | 0.24 | 0.27 | 0.76 | 0.76 | 0.73 | x | x | x | x | x | x |

Both variable elimination and Gibbs sampling produced accurate probability distributions across all three test networks. Variable elimination achieved exact matches with expected values in all test cases, validating the correctness of factor operations, evidence reduction, and elimination ordering. Gibbs sampling produced close approximations that converged toward exact values, with most probabilities matching within 0.01-0.02 of ground truth for the Child and Insurance networks, while Win95pts had additional challenges.

The Child network demonstrated strong agreement between both algorithms and expected values across all evidence conditions. Gibbs sampling deviated by at most 0.01 in individual states. The simplicity of the network allowed both algorithms to perform efficiently without significant challenges. The Insurance network exhibited equally strong performance.

The Win95pts network presented the most challenging test case due to its larger size and complexity. Variable elimination still produced exact results for all six problem variables across all scenarios. Gibbs sampling showed larger approximation errors under certain conditions. In particular, for problem 2 under P1 evidence, Gibbs sampling produced 0.05 versus the expected 0.08 for the Too_Long state. Problem 3 under P3 evidence showed 0.13

versus 0.26 expected for the No state. Problem 6 under P3 evidence exhibited the largest deviation at 0.14 versus 0.33 expected for the Yes state. These errors suggest that certain evidence configurations create sampling difficulties, possibly due to strong dependencies or near-deterministic relationships that slow convergence.

Despite the discrepencies on the Win95pts network, Gibbs sampling maintained overall accuracy across the majority of test cases. Most probability estimates fell within 0.02 of expected values, while many matched exactly.

## 5 Discussion

### 5.1 Challenges

Gibbs sampling initially did not work for the win95pts.bif file. The probabilities in this file are extremely deterministic, therefore, we had to introduce a smoothing factor to mellow out these probabilities. The smoothing factor makes sure that all probabilities lie between $(smoothing, 1 - smoothing)$; with a smoother factor equal to 0.001 all probabilities are between $(0.001, 0.999)$. This allows any state to be chosen, even if the probability is extremely minuscule.

Initially, variable elimination used a static ordering that sorted variables by their number of parents, in ascending order. This heuristic eliminated variables with fewer parents first, under the assumption that such variables would have smaller factors. Though the approach worked on the Child network, it was a failure for the Insurance network as the static ordering failed to account for how factors can grow during elimination. If the elimination order causes many variables to be combined at the same time, there can be exponentially many intermediate factors containing many variables.

### 5.2 Limitations

The performance dilemma for Gibbs sampling is noteworthy. As an approximate inference algorithm, increasing iteration count improves convergence but extends runtime proportionally. Our implementation uses fixed iteration counts scaled only by network size, without tuning for specific network topologies or evidence configurations.

Variable elimination faces a tradeoff between elimination ordering quality and compute. The greedy min-degree heuristic is good at computing orderings quickly but can generate larger intermediate factors, leading to potential failure on more complex networks. If more complex networks were evaluated, a more comprehensive approach could evaluate multiple different ordering strategies and select based on network-specific characteristics.

## 6 Summary

Both inference algorithms successfully computed posterior probabilities across all three test networks. Variable elimination's performance depends heavily on elimination ordering and network connectivity. The greedy min-degree heuristic proved sufficient for the test networks but may fail on larger or more densely connected network structures. Gibbs sampling demonstrates the tradeoff between compute cost and accuracy, with iteration count directly affecting both convergence quality and runtime.

There are several opportunities for improvement that exist. Variable elimination could benefit from more sophisticated ordering heuristics to handle larger networks. Gibbs sampling requires better parameter tuning strategies that adapt to network topology and evidence strength rather than relying solely on network size. Despite these limitations, both implementations proved functional and highlighted the advantages of exact versus approximate methods for probabilistic reasoning in Bayesian networks.

## 7 Appendix

**Jackson wrote the following:**

- Variable Elimination
- Paper - Abstract, 1.1, 1.2, 1.3, 2.1

**AJ wrote the following:**

- Gibbs Sampling
- Paper - 2.2, 3, 5.1, 5.2, 6, References

## References

[1] BNLearn. Bayesian Network Repository. Available at `http://www.bnlearn.com/bnrepository/`. Accessed: 2025-11-03.