A Bayesian Analysis of Simulation Algorithms for Inference in Belief Networks

Paul Dagum and Eric Horvitz
Section on Medical Informatics
Stanford University School of Medicine
Stanford University
Stanford, CA 94305-5479

Rockwell Palo Alto Laboratory
444 High Street
Palo Alto, CA 94301

January 18, 1993

Abstract

A belief network is a graphical representation of the underlying probabilistic relationships in a complex system. Belief networks have been employed as a representation of uncertain relationships in computer-based diagnostic systems. These diagnostic systems provide assistance by assigning likelihoods to alternative explanatory hypotheses in response to a set of findings or observations. Approximation algorithms have been used to compute likelihoods of hypotheses in large networks. We analyze the performance of leading Monte Carlo approximation algorithms for computing posterior probabilities in belief networks. The analysis differs from earlier attempts to characterize the behavior of simulation algorithms in our explicit use of Bayesian statistics: We update a probability distribution over target probabilities of interest with information from randomized trials. For real $\epsilon, \delta < 1$ and for a probabilistic inference $\Pr[x|e]$, the output of an inference approximation algorithm is an (ϵ, δ)-*estimate* of $\Pr[x|e]$ if with probability at least $1 - \delta$ the output is within relative error ϵ of $\Pr[x|e]$. We construct a stopping rule for the number of simulations required by *logic sampling*, *randomized approximation schemes*, and *likelihood weighting* to provide (ϵ, δ)-estimates of $\Pr[x|e]$. With probability $1 - \delta$, the stopping rule is *optimal* in the sense that the algorithm performs the minimum number of required simulations. We prove that our stopping rules are insensitive to the prior probability distribution on $\Pr[x|e]$.

1 Introduction

Belief networks are an expressive graphical language for representing uncertain knowledge about the causal and associational relationships among variables in complex systems. Over the last five years, there has been a maturation of techniques for assessing and solving belief networks (e.g., see *Networks*, 20, 1990). Several effective diagnostic reasoning systems use belief networks to assign probabilities to alternative hypotheses about a patient's health—for example, MUNIN [Andreassen et al., 1987], ALARM [Beinlich et al., 1989], Pathfinder [Heckerman et al., 1992], Sleep Consultant [Nino-Murcia and Shwe, 1991], VPnet [Rutledge et al., 1989], and QMR-DT [Shwe et al., 1991a]—or about the source of failure in complex machinery, including jet engines, electric power generators, and copy machines [Horvitz et al., 1988, Henrion et al., 1992, Breese et al., 1992]. Several exact inference algorithms have been developed for computing posterior probabilities with belief networks. Exact algorithms include the method of cutset conditioning, developed by Pearl [Pearl, 1988], and the clique-tree method, developed by Lauritzen and Spiegelhalter [Lauritzen and Spiegelhalter, 1988]. These exact inference methods exploit the topology of a belief network to compute the posterior probabilities of the belief network given the observed evidence.

Unfortunately, several complex knowledge bases that have been developed resist solution with exact inference algorithms. For example, the QMR-DT knowledge base [Shwe et al., 1991a], a reformulation of the Internist-1/QMR knowledge base [Miller et al., 1986] for internal medicine into a belief network, poses a difficult challenge to medical informatics investigators [Henrion, 1990]. Monte Carlo simulation procedures offer the most promising method for solving inference in this and similarly complex belief networks. Difficulties characterizing the convergence behavior of simulation algorithms hinders their use. For example, in the QMR-DT project, it is difficult to derive the confidence in simulation results as computation proceeds [Shwe and Cooper, 1991, Shwe et al., 1991b].

Simulation procedures have have been employed for solving diverse problems in computer science. To date, work on the characterization of simulation algorithms has been based largely on statistical methods, such as the Zero–One Estimation Theorem. From a Bayesian perspective, simulation-based inference techniques can be viewed as discarding useful information about a posterior probability of interest. However, such methods are satisfying to many in their ability to bypass complex distribution-updating procedures, and issues surrounding the use of prior probability. Nonetheless, we show how Bayesian updating can be used to characterize the convergence of simulation algorithms on results of interest by employing a conjugate distribution to simplify the updating procedure. Furthermore, we address the discomfort that some computer scientists and statisticians may have with methods that require prior probability distributions by proving that our stopping rules are insensitive in most cases to prior probabilities.

2 Belief Networks and Inference Algorithms

A belief network is a directed acyclic graph (DAG) containing nodes, representing propositions (*e.g.*, hypotheses and observations), and arcs representing probabilistic dependencies among nodes. Nodes representing propositions are associated with a set of mutually exclusive and exhaustive values that represent alternative possible states of a proposition (*e.g.*, true, false).

Let $\{X_1, ..., X_n\}$ denote the set of nodes in a belief network. For any node X_i, and set of parents π_{X_i}, the belief network specifies a conditional probability function $\Pr[X_i|\pi_{X_i}]$. The full joint-probability distribution specified by a belief network can be calculated by taking the product of the conditional probabilities,

$$\Pr[X_1, ..., X_n] = \prod_{i=1}^{n} \Pr[X_i|\pi_{X_i}]. \tag{1}$$

Probabilistic inference in belief networks refers to the computation of an *inference probability*—that is, $\Pr[\mathcal{X} = x|\mathcal{E} = e]$ for any given set of nodes $\mathcal{X}$ instantiated to value x and conditioned on observation nodes $\mathcal{E}$ instantiated to value e. Probabilistic inference in large multiply connected belief networks is difficult. Complexity analyses shows that both exact and approximate algorithms pose intractable problems in the worst case [Cooper, 1990, Dagum and Luby, 1991]. Nevertheless, for many problems, inference approximation procedures provide useful estimates of posterior probabilities in acceptable computation times.

The major classes of approximation algorithms for probabilistic inference in belief networks are *stochastic simulation* methods [Pearl, 1987b, Pearl, 1987a, Henrion, 1988, Fung and Chang, 1989, Shachter and Peot, 1989, Shachter and Peot, 1990, Chavez and Cooper, 1990] and *search-based* methods [Cooper, 1984, Henrion, 1990]. Stochastic simulation algorithms for probabilistic inference began with the pioneering work of Pearl's *straight simulation* algorithm

[Pearl, 1987b, Pearl, 1987a]. Simulation algorithms devised subsequently include *logic sampling* [Henrion, 1988], *likelihood weighting* [Shachter and Peot, 1989, Fung and Chang, 1989, Shachter and Peot, 1990], and *randomized approximation schemes* [Chavez and Cooper, 1990, Chavez, 1990, Dagum and Chavez, 1991]. We analyze stochastic-simulation algorithms in detail in Section 7. We distinguish straight simulation from other simulation algorithms because straight simulation is based on a *single long-run* rather than a *multiple short-runs* algorithm—that is, the estimate of straight simulation is based on a single, long simulation run rather than on the arithmetic mean of the outputs of multiple short runs. As we shall discuss in Section 8, this distinction has important consequences for the analyses of convergence of these algorithms.

3 Monte Carlo Methods

Stochastic simulation algorithms are based on the Monte Carlo method, a procedure for computing random variables based on a weighted random sampling. In this section, we discuss the foundations of the Monte Carlo method and simulation algorithms for probabilistic inference in belief networks, based on the Monte Carlo method. In Section 4 we discuss the rate of convergence of these algorithms.

The Monte Carlo method originated with the work of von Neumann and Ulam for computing parameters needed to construct nuclear reactors [von Neumann, 1951]. In this article we focus on the use of Monte Carlo for approximating Lebesgue integrals, or more specifically, for approximating expectations of random variables.

Let us first review the basics of Monte Carlo approximation of expectations of random variables. Let $(\Omega, 2^\Omega, \mathbf{P})$ denote a probability space, where 2^Ω denotes the power set of the set Ω and $\mathbf{P}$ denotes probability distribution. Let

$$\zeta : \Omega \rightarrow \Re$$

map a random variable into the reals $\Re$.

For a finite space Ω, the expectation $\mathbf{E}\zeta$ is the Lebesgue sum of the 2^Ω-measurable function $\zeta = \zeta(\omega)$ with respect to the Lebesgue measure $\mathbf{P}$,

$$\mathbf{E}\zeta = \sum_{\omega \in \Omega} \zeta(\omega)\mathbf{P}[\omega]. \tag{2}$$

For simplification, we shall use ϕ to denote $\mathbf{E}\zeta$. The Monte Carlo method approximates ϕ in Equation 2, by simulating the random variable ζ. The simulation of ζ requires being able to sample the space Ω with distribution given by the probability $\mathbf{P}$. An algorithm that accomplishes the latter is said to be a *sample generator* for the probability space $(\Omega, 2^\Omega, \mathbf{P})$. The output $\omega_0, \omega_1, \ldots$ of a sample generator for $(\Omega, 2^\Omega, \mathbf{P})$ defines the simulation of the random variable, $\zeta(\omega_0), \zeta(\omega_1), \ldots$. The arithmetic mean μ of a simulation of ζ of size N is an estimate of ϕ,

$$\mu = \frac{1}{N}(\zeta(\omega_1) + \cdots + \zeta(\omega_N)). \tag{3}$$

The Law of Large Numbers guarantees convergence of the estimate to ϕ in the limit of infinite samples,

$$\mu \to \phi \text{ as } N \to \infty.$$

Although the estimate μ has correct limiting convergence, from a computational perspective, only a finite number of samples are available. Thus, practical considerations dictate the need for lower bounds on the number of samples N required to bound the error in the estimate μ. The sequence $\zeta(w_0), \zeta(w_1), \ldots$ forms a sequence of independent and identically distributed random variables. Thus, by the Central Limit Theorem, for sufficiently large N the distribution of the estimate μ is approximated by a normal distribution. Under this assumption, the cumulative mass of the tails of the normal distribution gives the probability that μ does not approximate ϕ to within a specified error tolerance. Thus, in the limit as N tends to infinity,

$$\mathbf{P}\left[\frac{|\mu - \phi|}{\sqrt{\mathbf{V}\zeta}} \geq \epsilon\right] \to \frac{2}{\sqrt{2\pi}} \int_{-\infty}^{\epsilon} e^{-x^2/2} dx, \tag{4}$$

where $\mathbf{V}\zeta$ denotes the variance of the sequence $\zeta(w_0), \zeta(w_1), \ldots$.

Although the Central Limit Theorem guarantees that Equation 4 holds for infinite N, we do not know how well the approximation holds for finite N. This makes Equation 4 difficult to use in practice. Chebyshev's inequality bounds the probability of Equation 4 without any assumptions on the form of the probability distribution of the estimate μ. Chernoff bounds and the Zero-One Estimator Theorem (see, e.g., [Karp et al., 1989]) give bounds for Bernoulli trials, but they are expressed in terms of the unknown quantity ϕ.

4 Zero–One Convergence Analysis

The analysis of the convergence of simulation algorithms in the theoretical computer science community has been rooted in Zero–One Estimation Theory. The methodology has been used by several investigators to analyze the performance of simulation algorithms [Shachter and Peot, 1989, Chavez and Cooper, 1990, Dagum and Chavez, 1991].

Zero–One Estimation Theory provides a lower bound, N, on the number of Bernoulli trials required to achieve a specified level of accuracy in the estimate of the inference. This approach is deficient in that N is expressed in terms of the probabilistic inference to be computed, and thus, it requires *a priori* knowledge of the value of the inference. This problem is circumvented by using lower-bound estimates on the value of the inference, yielding upper bound estimates on N. Typically, however, the best *a priori* lower bounds one can achieve are exponentially small in the size of the problem, independent of the actual value of the inference. Thus, convergence analysis based on Zero–One Estimation Theory may suggest intractable bounds on the number of trials required even though a small, polynomial, number of trials may suffice.

We review briefly the Zero–One Estimation approach for analyzing the convergence of simulation algorithms. Consider the problem of trying to estimate $\mathbf{P}[\mathcal{A}]$ in the probability space $(\Omega, 2^\Omega, \mathbf{P})$ where $\mathcal{A}$ denotes a subset of Ω. From the presentation of the Monte Carlo method in Section 3, this problem can be solved by simulating the random variable $\zeta = \zeta(\omega)$ that is the characteristic function of the set $\mathcal{A}$. It is straightforward to show that $\phi = \mathbf{P}[\mathcal{A}]$. Let a *success* be defined as having chosen an element ω that belongs to $\mathcal{A}$—that is, $\zeta(\omega) = 1$. From the Law of Large Numbers, in the limit of an infinite number of trials, the fraction of successes, μ, converges to ϕ. If we stop a Monte Carlo simulation after a finite number of trials N, the current fraction μ serves as an estimate of ϕ.

For simulation algorithms, one is interested in deriving an upper bound on N which guarantees that μ gives a good estimate of ϕ. More specifically, for any $\epsilon, \delta \leq 1$ we would like to know the least number of trials N needed to guarantee that with a probability of at least $1 - \delta$, μ approximates ϕ with relative error ϵ.

Definition 4.1 *For $\epsilon, \delta \leq 1$, μ is an (ϵ, δ)-estimate of ϕ if and only if*

$$\mathbf{P}[\phi(1+\epsilon)^{-1} \leq \mu \leq \phi(1+\epsilon)] \geq 1 - \delta.$$

The Zero–One Estimator Theorem gives the following result.

Theorem 4.2 *If* $\epsilon, \delta \leq 1$ *and*

$$N = \frac{4}{\phi\epsilon^2} \ln \frac{2}{\delta}$$

then μ *is an* (ϵ, δ)*-estimate of* ϕ.

PROOF See [Karp et al., 1989]. □

Note that the upper bound on N given by the Zero–One Estimator Theorem is contingent on knowing ϕ—the same quantity that we are employing the simulation algorithm to estimate. To circumvent this circular definition, a conservative lower-bound on ϕ, computed with a polynomial-time algorithm, is used to derive a rough estimate of N. A key challenge for computer scientists is to develop methods for computing a lower bound on ϕ that is within a constant multiplicative factor of ϕ. Unfortunately, in many cases the best lower bound that can be computed is exponentially small in the size of the problem. Thus, traditional analyses on convergence, and associated recommendations about run times for achieving desired levels of confidence, can be quite conservative, lead to wasteful computations.

5 Bayesian Analysis of Convergence

We explore the relationship between Zero–One Estimation Theory and a Bayesian probabilistic analysis that exploits the structure of the second-order distributions over ϕ. We use *second-order* distribution to refer to a probability distribution over a probabilistic parameter of interest. Few analyses of belief-network inference algorithms have characterized the error of an estimate with a probability distribution on a target probability. Some algorithms output *partial* characterizations of such *second-order* distributions. For example, search- and decomposition-based procedures typically provide upper and lower bounds on a probability, without a statement of the likelihood of alternate values of the final answer within these bounds [Cooper, 1984, Henrion, 1990, Horvitz et al., 1989]. Also, Zero–One Estimation Theory can be viewed as relying on an incomplete characterization of the second-order distributions associated with a Bernoulli process. Rather than considering second-order probability distributions explicitly, investigators have resorted to conservative worst-case analyses of convergence. The variance of the outcomes of the simulation provides error bounds on the estimates. We prove that from a Bayesian perspective, previous analyses of the performance of simulation algorithms are conservative.

5.1 Overview

We employ the properties of a conjugate second-order probability distribution to develop a stopping rule for the number of Bernoulli trials required by a Monte Carlo algorithm to output an (ϵ, δ)-estimate. A *conjugate* distribution is a parameterized description of a probability distribution whose basic structure is invariant to Bayesian updating in response to new information. Conjugate distributions are valuable for simplifying the analysis of how a second-order distribution changes with the results of Monte Carlo simulation. We shall examine the beta density function, a conjugate probability distribution over possible means ϕ of a Bernoulli process. The stopping rule we develop considers the current estimate of ϕ to determine whether the estimate is within predefined error bounds, or to indicate that further computation is needed. With probability close to one—that is, $1 - 2^{-n}$ for problem size n—the algorithm terminates after the minimum number of trials. In Section 8, we apply the results of this section to analyze the convergence of simulation-based inference approximation algorithms.

Let μ be the estimate of $\Pr[x|e]$ provided by a stochastic-simulation algorithm after N simulations. We use Bayesian statistics to reason about the convergence of μ to $\Pr[x|e]$ as N increases. We update the state of information about a target inference, $\Pr[x|e]$, by employing a beta distribution with parameters μN and N. We sum the tails of the beta distribution to obtain an expression for the error of the estimate that is a function of μ and N. From this expression, we derive a stopping rule for the number of simulations which guarantees with probability $1 - 2^{-n}$ that the output estimates the inference with prescribed relative error ϵ. With probability $1 - 2^{-n}$, the stopping rule is optimal in the sense that the algorithm terminates after the minimum number of simulations.

5.2 Bayesian Updating with a Conjugate Distribution

For N trials and μN successes, the probability distribution over ϕ is described by a beta function as follows:

$$\beta(\phi|\mu, N) \quad = \quad B(\mu, N)\phi^{\mu N-1}(1-\phi)^{(1-\mu)N-1}$$

where

$$B(\mu, N) = \frac{(N-1)!}{(\mu N-1)!((1-\mu)N-1)!}.$$

The beta distribution has a mean of μ and variance of

$$v_\beta(\phi, N) \quad = \quad \frac{1}{N+1}\mu\,(1-\mu)\,.$$

If ϕ has a beta prior, then the probability distribution for ϕ is also a beta distribution after we observe N outcomes with μN successes [Howard, 1970]. In other words, the probability that a particular value of ϕ gives rise to a particular fraction of successes μ in the N Bernoulli trials is given by $\beta(\phi|\mu, N)$. Later, we will describe why our analyses of convergence are insensitive to the use of alternate information-poor prior probability distributions.

Theorem 5.1 *Let* $\Pr[\phi]$ *denote the prior distribution on* ϕ*. The distribution of* ϕ *after we observe* μN *successes in* N *Bernoulli trials is*

$$\Pr[\phi|\mu, N] = k\phi^{\mu N}(1-\phi)^{(1-\mu)N}\Pr[\phi],$$

where k *normalizes the distribution.*

Proof Let $\Pr[\phi|\mu, N]$ denote the distribution of ϕ having observed μN successes in N Bernoulli trials. We would like to know how this probability is updated when the outcome of the $N+1$st trial is made available. Let us first assume that the $N+1$st trial is a success, denoted by S. Bayes' theorem expresses the updated distribution of ϕ as

$$\Pr[\phi|\mu N+1, N+1] = \frac{\Pr[S|\phi, \mu N, N]\Pr[\phi|\mu N, N]}{\Pr[S|\mu N, N+1]}. \tag{5}$$

The probability we observe a success given knowledge of ϕ is just ϕ; thus

$$\Pr[S|\phi, \mu N, N] = \phi.$$

The denominator in Equation 5 does not depend on ϕ. Thus,

$$\Pr[\phi|\mu N+1, N+1] = k\phi \Pr[\phi|\mu N, N], \tag{6}$$

with normalization constant k. When the $N+1$st trial is a failure, a similar argument gives the expression

$$\Pr[\phi|\mu N, N+1] = k(1-\phi) \Pr[\phi|\mu N, N]. \tag{7}$$

The expression for the probability distribution of ϕ conditioned on observing μN successes in N trials can be simplified by recursive application of Equations 6 and 7. Thus,

$$\Pr[\phi|\mu N, N] = k\phi^{\mu N}(1-\phi)^{(1-\mu)N} \Pr[\phi], \tag{8}$$

where as before, k normalizes the distribution. □

5.3 Issues of Prior Probability

The explicit consideration of prior probability distributions in Bayesian reasoning has been a source of ongoing discussion among advocates of statistical and Bayesian methodologies. The results of Bayesian analysis of Monte Carlo simulation algorithms are insensitive to changes in assumptions about the specific information-poor prior probability distribution selected.

Consider the knowledge that an agent has about ϕ prior to observing the first sample. Before experimentation, an agent might believe that all values of ϕ in the interval $[0,1]$ are equiprobable. We represent this prior distribution with a uniform distribution, or, equivalently, with $\beta(\phi|\frac{1}{2}, 2)$. Alternatively, Bayesian statisticians use Jeffreys prior—the beta distribution $\beta(\phi|\frac{1}{2}, 1)$—for the unbiased-beta prior (see, e.g., [Hartigan, 1983]). This distribution effectively partitions its mass equally at the two extremes $\phi = 0$ and $\phi = 1$, reflecting the uncertainty in ϕ.

The analysis of the preferred form of the informationless beta prior is rendered immaterial by noting the general insensitivity of results to different information-poor prior probability distributions. For example, the information necessary to update an agent's prior distribution on ϕ, from Jeffreys prior to the uniform distribution, is provided by the outcome of a single sample. Thus, for large samples, the rate of convergence of the estimate to the mean is insensitive to the choice of the beta prior distribution on ϕ. Others have explored in detail the insensitivity of probabilistic analyses to the assumption of alternative prior distributions [DeRobertis and Hartigan, 1981, Hartigan, 1983].

From Theorem 5.1, if the prior distribution $\Pr[\phi]$ is beta, then the posterior distribution on the mean ϕ is also beta, reflecting the conjugate property of the beta distribution. We assume ϕ has the beta prior $\beta(\phi|\frac{1}{2}, 0)$. Like the Jeffreys prior, $\beta(\phi|\frac{1}{2}, 0)$ distributes its mass at the extremes $\phi = 0$ and $\phi = 1$, and thus, it is an informationless prior (see, e.g., [Hartigan, 1983]). We prove Corollary 5.2.

Corollary 5.2 *Prior to any observations, if the mean ϕ of a Bernoulli process is distributed according to the prior $\beta(\phi|\frac{1}{2}, 0)$, the distribution of ϕ after we observe μN successes in N trials is $\beta(\phi|\mu, N)$.* □

In Part II of the Appendix, we develop stopping rules for arbitrary prior probability distributions. We also demonstrate the insensitivity of convergence results to alternate prior probability distributions.

5.4 Bayesian Stopping Rules

We now develop stopping rules for simulation algorithms based on a Bayesian error analysis [Dagum and Horvitz, 1991]. To determine the least number of trials N required to output (ϵ, δ)-estimate, we use the beta distribution and the equivalence of Definition 4.1 to

$$\Pr\left[\mu(1+\epsilon)^{-1} \leq \phi \leq \mu(1+\epsilon)\right] > 1 - \delta. \tag{9}$$

With the unbiased-beta prior, Corollary 5.2 shows ϕ has a $\beta(\phi|\mu, N)$ distribution after we observe N outcomes with μN successes.

The cumulative mass of $\beta(\phi|\mu, N)$ that lies inside the bounds $[\mu(1+\epsilon)^{-1}, \mu(1+\epsilon)]$ yields the probability term in Equation 9. Conversely,

$$\int_0^{\frac{\mu}{(1+\epsilon)}} \beta(x|\mu, N)dx + \int_{\mu(1+\epsilon)}^1 \beta(x|\mu, N)dx = \delta. \tag{10}$$

yields the failure probability δ.

We can refer to beta cumulative distribution tables for necessary values. Alternatively, we can solve Equation 10 directly. Gauss' hypergeometric functions [Gradshteyn and Ryzhik, 1980] gives a closed-form solution of Equation 10,

$$\int_0^z \beta(x|\mu, N)dx = \frac{z^{\mu N}}{\mu N} F(\mu N, 1-(1-\mu)N; \mu N+1, z)B(\mu, N), \tag{11}$$

The hypergeometric function $F(\mu N, 1-(1-\mu)N; \mu N+1, z)$ is a polynomial in z of degree $(1-\mu)N-1$ and ith coefficient

$$(-1)^i \frac{\mu N}{\mu N + i}\binom{(1-\mu)N-1}{i}. \tag{12}$$

For arbitrary values of μ, N, and ϵ, Equation 11 solves the first integral in Equation 10. From the symmetry properties of the beta distribution, we express the second integral in Equation 10 as

$$\int_{\mu(1+\epsilon)}^{1} \beta(x|\mu, N)dx = \int_{0}^{1-\mu(1+\epsilon)} \beta(x|1-\mu, N)dx \tag{13}$$

and once again, Equation 11 gives a solution.

From the preceding analysis, for arbitrary values of μ, N, and $\epsilon \leq 1$, the sum of two hypergeometric functions yields the failure probability δ. Thus, we prove the following stopping rule for simulation algorithms.

Theorem 5.3 *Let ϕ have a beta-prior distribution. For input parameters $\epsilon, \delta \leq 1$, and current estimate μ, the approximation algorithm stops if δ equals the sum*

$$\left[\frac{a^{\mu N}}{\mu N} F(\mu N, 1-(1-\mu)N; \mu N+1, a) + \frac{b^{\overline{\mu} N}}{\overline{\mu} N} F(\overline{\mu} N, 1-\mu N; \overline{\mu} N+1, b)\right] B(\mu, N),$$

where $a = \mu(1+\epsilon)^{-1}$, $b = 1-\mu(1+\epsilon)$, and $1-\mu$ denotes $\overline{\mu}$. □

We compute Gauss's hypergeometric functions to yield the error δ of the estimate μ in Theorem 5.3. This computation is expensive. We could base our stopping rule on a more inexpensive table lookup. However, we can also approximate the analysis. In Section 6, we derive an approximation to δ that we compute in constant time. Thus, we obtain a stopping rule that is evaluated in constant time.

6 Traditional Versus Bayesian Analyses of Run Time

We now derive upper bounds on the number of trials N for a stochastic simulation algorithm to output an (ϵ, δ)-estimate. Theorem 4.2, the Zero-One Estimator Theorem, gives the number of trials for a stochastic simulation algorithms to output an an (ϵ, δ)-estimate in terms of ϕ—that is, the limit value of μ for infinite trials. We do not know ϕ and Theorem 4.2

provides only a *theoretical* measure of the efficiency of simulation algorithms. In practice, a lower bound for ϕ is used in Theorem 4.2 to get an upper bound on the number of trials N. Thus, the *practical* efficiency of simulation approximation algorithms rest critically on the nature of the available lower bound for ϕ.

Instead of relying on Gauss's hypergeometric functions to compute δ, we approximate the tails of the beta distribution by a Taylor series and estimate the cumulative mass of Equation 10 by a Reimann sum. In this way, we obtain an easily computable expression for N in terms of μ, ϵ, and δ.

Theorem 6.1 *Let ϕ have a beta-prior distribution. For given $\epsilon, \delta \leq 1$, and estimate μ, if N satisfies*

$$N > \frac{7}{\epsilon^2 \mu} \ln \left(\frac{4}{\delta} \right)$$

then μ is an (ϵ, δ)-estimate.

Proof See Appendix, Part I. □

Theorem 6.1 replaces the appearance of the unknown mean ϕ in the Zero-One Estimator Theorem with the *known* current estimate μ. Thus, Theorem 6.1 provides a practical bound on the required number of trials dervied from a principled analysis of information made available about ϕ via simulation.

Corollary 6.2 *An approximation algorithm outputs an (ϵ, δ)-estimate after N simulations, where N satisfies*

$$\mathbf{P} \left[N \leq \frac{14}{\epsilon^2 \phi} \ln \left(\frac{4}{\delta} \right) \right] \geq 1 - \delta.$$

Proof The algorithm stops when N satisfies Theorem 6.1. Thus, μ is an (ϵ, δ)-estimate. Therefore, when the algorithm stops, with probability at least $1 - \delta$, we know $\phi \leq 2\mu$. We substitute $\phi/2$ in Theorem 6.1 to yield an upper bound on N in terms of ϕ that holds with probability $1 - \delta$. □

The upper bound for N in Corollary 6.2 is within a multiplicative constant of the lower bound, and therefore, it is optimal. Thus, with probability at least $1 - \delta$, the stopping rule expressed by Theorem 6.1 is optimal. With probability δ, the output is not an (ϵ, δ)-estimate, in which case Corollary 6.2 does not hold and the stopping rule is not optimal.

By Theorem 6.1, if the failure probability δ is $O(2^{-n})$ then the number of trials increases by a factor of $O(n)$. If we accept an $O(n)$ increase in N then the failure probability is made very small. Since the test condition expressed by the stopping rule is easily implemented, by Corollary 6.2 with high probability we achieve optimal convergence of simulation algorithms for probabilistic inference.

Theorem 5.3 and Theorem 6.1 express stopping rules contingent on a beta-prior distribution of the estimate μ. In Section 5.3, we discussed the insensitivity of the distribution for μ to beta priors once we observe several outcomes. In Part II of the Appendix, we derive stopping rules for situations where the prior probability is not a beta distribution.

REMARK Thus far, we have discussed Monte Carlo algorithms that ouput *randomized-relative error* approximations. That is, the estimate μ satisfies Definition 4.1. We can weaken the definition of randomized-relative error approximations to yield *randomized-absolute error* approximations. For $\epsilon, \delta \leq 1$, μ is a randomized-absolute error approximation if and only if

$$\mathbf{P}[\phi - \epsilon \leq \mu \leq \phi + \epsilon] \geq 1 - \delta.$$

If instead of randomized-relative error approximations we desire randomized-absolute error approximations, then

$$N > \frac{2}{\epsilon^2} \ln \frac{1}{\delta}$$

trials suffice for convergence. The result follows directly from the analysis of the Zero-One Estimator Theorem. Although we can derive a similar result from the analysis presented in Part I of the Appendix, it is unnecessary to invoke a Bayesian analysis to derive a stopping rule for absolute error approximations. For absolute-error approximations, N is independent of ϕ and implementation of the bound is straightforward.

7 Simulation Algorithms for Inference

We describe logic sampling, straight simulation, randomized approximation schemes, and likelihood-weighting simulation algorithms for inference in belief networks. In Section 8, we employ results from Section 5 to analyze the specific behavior of these algorithms.

Let the nodes of the belief network be denoted by $\mathcal{W} = \{W_1, ..., W_n\}$, the unobserved nodes by $\mathcal{Z} = \{Z_1, ..., Z_m\}$, and as before we denote the observed nodes by $\mathcal{E} = \{E_1, ..., E_q\}$, and $\mathcal{X} = \{X_1, ..., X_p\}$ denotes some subset of $\mathcal{Z}$. Lower-case letters, $w = (w_1, ..., w_n)$, $z = (z_1, ..., z_m)$, $e = (e_1, ..., e_q)$ and $x = (x_1, ..., x_p)$, denote instantiations of these nodes. We also refer to special types of nodes, based on the topology of a belief network. *Root* nodes are those nodes that have no parents.

We define logic sampling, straight simulation, randomized approximation schemes, and likelihood weighting in terms of the Monte Carlo method presented in Section 3. Each algorithm is defined in terms of (1) the probability space from which samples are generated; and, (2) the algorithm that generates samples from that space with the desired probability. In the context of this information, we define, for each algorithm, a random variable $\zeta = \zeta(\omega)$ with expectation equal to the desired inference probability.

7.1 Logic Sampling

We describe two implementations of logic sampling to highlight two basic Monte Carlo methods to approximating the inference, $\Pr[x|e]$. Both implementations have similar running times. In Sections 7.3 and 7.4 we show that randomized approximation schemes and likelihood weighting represent two versions of the basic methods discussed here. We begin with the sample-generator algorithm that generates complete instantiations of the belief network. This algorithm is common to both implementations of logic sampling.

The *sample-generator algorithm* begins simulating the root nodes in the belief network. For example, if w_i is a root node with prior probability $\Pr[w_i = 1] = p$, then the algorithm tosses a coin having probability p of landing heads. The algorithm instantiates w_i to 1 if the outcome is heads and instantiates w_i to 0 otherwise. Once all the root nodes are instantiated, the algorithm proceeds to the nodes in the network that have root nodes as their only parents. If w_j is such a node, having root node parents π_{w_j}, then the outcome of a coin toss with probability $\Pr[w_j = 1|\pi_{w_j}]$ of landing heads determines the setting of w_j. The procedure continues in this manner until all nodes have been instantiated. By this construction, instantiation $w = (w_1, ..., w_n)$ occurs with probability $\prod_{i=1}^{n} \Pr[w_i|\pi_{w_i}]$. It follows from Equation 1 that the probability of the outcome w is drawn with the probability $\Pr[w]$.

7.1.1 Basic Method

In the basic method, the probability space has state space $\Omega = \{w : w = (w_1, ..., w_n), w_i = 0, 1\}$ and probability distribution given by the full joint distribution of the belief network, $\mathbf{P} = \Pr$. The basic method makes one call to the sample-generator algorithm to output $w \in \Omega$ with probability distribution $\Pr[w]$. To compute inferences conditional on evidence e—that is, $\Pr[x|e]$—the basic method obtains estimates of the unconditional inferences $\Pr[x, e]$ and $\Pr[e]$ separately. The ratio of the unconditional inferences gives an estimate of $\Pr[x|e]$. We obtain estimates of $\Pr[e]$ and $\Pr[x, e]$ simultaneously from the outcomes of each simulation. For example, to estimate $\Pr[x, e]$, logic sampling defines a random variable $\zeta = \zeta(w)$ that takes on the value 1 if in the instantiation w, nodes $\mathcal{X}$ and $\mathcal{E}$ are instantiated to x and e, and takes on the value 0 otherwise. It is readily verified that with this definition, ζ has the required expected value

$$\phi = \Pr[x, e].$$

7.1.2 Rejection Method

The rejection method estimates the probability $\Pr[x|e]$ directly by accepting outcomes only if nodes $\mathcal{E}$ are instantiated to e. The rejection method converges faster on the estimate. This property follows directly from Theorem 4.2 or from the Bayesian analysis in Section 8. The disadvantage of the rejection method is the increase in running time to generate valid outcomes, since multiple calls are made to the sample-generator algorithm.

The rejection method defines a random variable $\zeta = \zeta(w)$ to take on the value 1 if, in the instantiation w, nodes $\mathcal{X}$ and $\mathcal{E}$ are instantiated to x and e, and to take on the value 0 if nodes $\mathcal{X}$ are not instantiated to x *but* nodes $\mathcal{E}$ *are* instantiated to e. Outcomes for which nodes $\mathcal{E}$ are not instantiated to e are rejected. With this definition, ζ has expected value

$$\phi = \Pr[x|e].$$

The expected number of simulations required to generate a valid outcome—that is, one for which $\mathcal{E}$ is instantiated to e—is $\frac{1}{\Pr[e]}$.

The two implementations of logic sampling differ in the time T to generate a sample. In the basic method, T is $O(n)$—that is, the time to run the sample-generator algorithm. In the rejection method, T is a random variable with expected value $\mathbf{E}T = O(\frac{n}{\Pr[e]})$.

7.2 Straight Simulation

In straight simulation, the probability space has state space Ω that consists of all possible instantiations of the m unobserved nodes $\mathcal{Z}$. Instantiations are generated by a sequence of random variables $\xi_0, \xi_1, ...,$ with values in Ω, that form a stationary Markov process. Let $\Omega = \{w_0, ..., w_{2^m-1}\}$ denote an arbitrarily fixed ordering of the states. The transition probability p_{ij} of going from state ω_i to state ω_j of the Markov process is defined as follows. Assume ω_i and ω_j differ in the instantiation of a single node Z_i, which is instantiated to z_i in ω_i and $\overline{z}_i = 1 - z_i$ in ω_j. Let w_{Z_i} denote the instantiation of all nodes in $\mathcal{Z}$ except node Z_i to ω_i. Then, define

$$p_{ij} = \Pr[\overline{z}_i | w_{Z_i}, e]. \tag{14}$$

If ω_i and ω_j differ in more than a single node, then $p_{ij} = 0$.

Straight simulation fixes an arbitrary order $Z_1, ..., Z_m$ of the nodes in $\mathcal{Z}$. The algorithm cycles through the nodes in order, and, at each node, the algorithm simulates the conditional distribution given by Equation 14 for that node. The algorithm instantiates the node to the outcome of the simulation. The conditional distribution in Equation 14 can be simulated in $O(n)$ time because it is shown that the conditional distribution can be decomposed into a product of the conditional probabilities that define the belief network [Pearl, 1987b].

The Markov process is stationary with distribution $\{\Pr[w|e] : w \in \Omega\}$ [Pearl, 1987b]. Let $(p_i)_0^{2^m-1}$ denote the stationary distribution and let $p_{ij}^{(t)}$ denote the tth order transition probabilities. Because the transition matrix (p_{ij}) is irreducible and aperiodic, then

$$\lim_{t \to \infty} p_{ij}^{(t)} \to p_j. \tag{15}$$

To compute the inference probability $\Pr[x|e]$, straight simulation defines a random variable $\zeta = \zeta(\omega)$ that takes on the value 1 if in instantiation ω nodes $\mathcal{X} \subset \mathcal{Z}$ are instantiated to x, and it takes on the value 0 otherwise. If $\omega^0, \omega^1, ...$ denotes the outcomes of straight simulation, then

$$\mu = \sum_{i=0}^{t} \zeta(\omega^i) \tag{16}$$

converges to $\phi = \Pr[x|e]$ in the limit $t \to \infty$. The convergence depends on the convergence expressed in Equation 15. The latter forms a ϕ-mixing process [Billingsley, 1968] that we now analyze.

We assume that the conditional probabilities that define the belief network are nonzero. Thus, $p_i > 0$ for all i, and

$$\phi_t = \max_{ij} \left| \frac{p_{ij}^{(t)}}{p_j} - 1 \right| \tag{17}$$

is finite. The sequence $\phi_0, \phi_1, \ldots$ measures the convergence rate of the Markov process to the stationary distribution (also known as the mixing rate of the Markov process). The transition matrix (p_{ij}) is nonnegative, and Perron-Frobenius theory dictates that the second eigenvalue λ_1 of the transition matrix bounds ϕ_n from above [Seneta, 1973],

$$\phi_n \leq \lambda_1^n.$$

Evaluation of λ_1 is inaccessible to direct computation because of the size of the transition matrix. It is desirable, however, to have an upper bound on λ_1 to inform us when we have done sufficient simulations so that ϕ_n is less than a prescribed threshold. The *conductance* of the Markov process gives an upper bound on λ_1 [Sinclair and Jerrum, 1989]. With the conductance, we establish the number of simulations T for which ϕ_T is $O(2^{-n})$. From Equation 17 we verify that after T simulations, the Markov process is in state ω with probability p,

$$\Pr[\omega|e](1 - \phi_T) \leq p \leq \Pr[\omega|e](1 + \phi_T).$$

Thus, the outcomes $\omega^T, \omega^{2T}, \ldots$ are approximately independently and identically distributed with probability distribution that approximates $\Pr[\omega|e]$. This result forms the basis of randomized approximation schemes discussed in the section that follows.

7.3 Randomized Approximation Schemes

Pearl's straight simulation algorithm is an example of a single long-run approximation algorithm. Logic sampling, randomized approximation schemes, and likelihood weighting are examples of multiple short-run approximation algorithms. Like straight simulation, the latter algorithms also simulate conditional distributions. Unlike straight simulation, however, these algorithms run multiple short simulations and output the arithmetic mean of the estimates of each simulation. If straight simulation is implemented with multiple short runs then we have a randomized approximation scheme. In a randomized approximation scheme the cycle order through the nodes is randomized to simplify the analysis of convergence.

The choice between using a single long-run or a multiple short-run simulation algorithm is unsettled. Efficiency is the advantage of the single long-run algorithm [Raftery and Lewis, 1992]. In practice, however, the accuracy of the output of a single long-run simulation is difficult to assess because we do not know when a single run has converged. For this reason, others have argued that it is important to use multiple short runs [Gelman and Rubin, 1991].

A randomized approximation scheme is a many short-runs version of straight simulation. When we sample every Tth simulation only, for sufficiently large T, the outcomes of straight simulation are approximately independently and identically distributed with probability distribution that approximates $\Pr[\omega|e]$. To estimate the inference $\Pr[x|e]$, we define the random variable $\zeta = \zeta(\omega)$ that takes on the value 1 if in the instantiation ω, nodes $\mathcal{X}$ are instantiated to x, and it takes on the value 0 otherwise. From this definition and Equation 17 we verify that $\zeta(\omega^T), \zeta(\omega^{2T}), \ldots$ is a Bernoulli process with probability p that satisfies

$$\Pr[x|e](1-\phi_T) \le p \le \Pr[x|e](1+\phi_T). \tag{18}$$

The estimate μ is defined by

$$\mu = \sum_{i=1}^{N} \zeta(\omega^{iT}).$$

In the limit of infinite N, μ converges to p rather than $\phi = \Pr[x|e]$. However, for sufficiently large T, this difference is insignificant in contrast to the error in the estimate μ that arises because the number of samples we use for the estimate is finite [Broder, 1986, Jerrum et al., 1986].

In a randomized approximation scheme the runs are of length T and N is the number of runs. When

$$\phi_T < \frac{1}{3}\epsilon \Pr[x|e], \tag{19}$$

the relative error ϕ_T in Equation 18 is comparable to the relative error ϵ of the estimate μ [Broder, 1986, Jerrum et al., 1986]. Dagum and Chavez [Dagum and Chavez, 1991] obtain an upper bound on T so that ϕ_T satisfies Equation 19 by application of nonasymptotic results on the rate of convergence of ergodic Markov chains [Sinclair and Jerrum, 1989].

Once we have a value for T, the Zero–One Estimator Theorem provides a bound on the number of simulations N so that with probability at least $1-\delta$, the output estimates $\Pr[x|e]$ with relative error ϵ.

7.4 Likelihood Weighting

Likelihood weighting employs the same probability state space Ω that does a randomized approximation algorithm. The probability distribution over the state space is the *sampling distribution* ρ; thus, $\mathbf{P} = \rho$. Each variation of likelihood weighting employed defines a specific sampling distribution. Variations of the likelihood weighting algorithm include the *basic method*, several forms of *importance sampling* and *Markov-blanket scoring* [Shachter and Peot, 1989].

7.4.1 Basic Method

In the *basic method*, the sampling distribution is the *path probability*,

$$\rho(z, e) = \prod_{z_i \in z} \Pr[z_i | \pi_{z_i}]|_{\mathcal{E}=e}.$$

Sampling the space Ω according to the path probability is accomplished by a simulation procedure similar to logic sampling, except that only the nodes in $\mathcal{Z}$ are simulated. It is evident that the probability of outcomes of the sampling distribution differs from the full joint probability of the belief network. In likelihood weighting, if an estimate weighs each outcome equally—as is done, for example, in logic sampling and randomized approximation schemes—then the output is biased. Instead, in likelihood weighting the sampling distribution determines a *weighting distribution* $\omega(z, e)$ defined by

$$\omega(z, e) = \frac{\Pr[z, e]}{\rho(z, e)}. \tag{20}$$

Thus, likelihood weighting scores the weights $\omega(z, e)$ for each outcome z to yield unbiased estimates.

As in logic sampling, likelihood weighting estimates $\Pr[x, e]$ and $\Pr[e]$ to obtain an estimate of the inference probability $\Pr[x|e]$. To estimate $\Pr[x, e]$, likelihood weighting defines a random variable $\chi = \chi(z)$ to take on the value 1 if in the instantiation z nodes $\mathcal{X}$ are instantiated to x, and to take on the value 0 otherwise. Likelihood weighting then scores the random variable ζ,

$$\zeta(z) = \chi(z) \cdot \omega(z, e).$$

We verify that $\mathbf{E}\zeta = \Pr[x, e]$. To estimate $\Pr[e]$, likelihood weighting scores the random variable ξ,

$$\xi(z) = \omega(z, e).$$

We verify that $\mathbf{E}\xi = \Pr[e]$.

In the basic method, the interval $[0, 1]$ contains all the values of $\omega(z, e)$. Thus, the convergence of a $0 - 1$ Bernoulli process upper bounds the convergence of values of the random variables ζ and ξ.

7.4.2 Importance-Sampling Method

Importance sampling is an established method of reducing the standard error in the estimate of Monte Carlo algorithms [Rubinstein, 1981]. For example, the basic algorithm estimates the probability $\Pr[e]$ by sampling the path probability $\rho(z, e)$

$$\Pr[e] = \sum_z \omega(z, e)\rho(z, e).$$

The distribution $\rho(z, e)$ is not necessarily the best choice of sampling distribution to use in a Monte Carlo approximation for $\Pr[e]$. Importance sampling introduces a different distribution $\tilde{\rho}(z, e)$ in the approximation of $\Pr[e]$,

$$\Pr[e] = \sum_z \left[\frac{\omega(z, e)\rho(z, e)}{\tilde{\rho}(z, e)}\right] \tilde{\rho}(z, e). \tag{21}$$

The sampling distribution $\tilde{\rho}(z, e)$ defines the new weighting distribution $\tilde{\omega}(z, e)$ given by the first term in the sum of Equation 21. The choice of distribution that minimizes the variance of a Monte Carlo estimate is

$$\tilde{\rho}(z, e) = \frac{\omega(z, e)\rho(z, e)}{\Pr[e]}. \tag{22}$$

Because $\omega(z, e)\rho(z, e) = \Pr[z, e]$, Equation 22 yields the distribution $\Pr[z|e]$.

Unfortunately, we cannot simulate easily the optimal choice of distribution. The rejection method of logic sampling and randomized approximation schemes sample the distribution $\Pr[z|e]$ but at the expense of a long running time for each sample. For variance reduction, however, it suffices to choose a distribution with shape similar to the distribution in Equation 22 [Rubinstein, 1981].

By sampling with a distribution $\tilde{\rho}$ that approximates $\Pr[z|e]$, importance sampling requires fewer samples to estimate $\Pr[e]$ than the basic method of logic sampling. Only when the algorithm samples $\tilde{\rho}$ efficiently, is the savings in samples beneficial. The rejection method of logic sampling and randomized approximation schemes sample the distribution $\Pr[z|e]$ at a substantial increase in the running time. The savings in running time of these algorithms comes from the reduction in the number of samples to estimate $\Pr[x|e]$ compared to $\Pr[x, e]$ and $\Pr[e]$.

8 Simulation Performance Analyses

To analyze the performance of different simulation algorithms, we must consider (1) the efficiency with which valid, or usable, samples are generated, and (2) the complexity of the informational update associated with each valid sample. We focus on the sample generation and information update for logic sampling, randomized approximation, and likelihood weighting methods.

As we noted in Section 4, investigators have previously analyzed the performance of stochastic simulation algorithms with the Zero–One Estimation Theorem [Chavez and Cooper, 1990], or with use of weaker bounds such as Chebyshev's inequality [Shachter and Peot, 1989]. These theorems are used as key arguments in proofs that bound the number of samples N required by an (ϵ, δ)-estimate. Since these bounds require a priori knowledge of the inference probability being estimated, they are only of theoretical interest—that is, they cannot be employed to appropriately limit computation in real-world implementations.

The bounds used in practice overestimate N. For example, to estimate $\Pr[x|e]$ for a single node instantiation x, Chavez and Cooper [Chavez and Cooper, 1990] use the smallest Markov blanket probability for x as the lower bound of $\Pr[x|e]$. The lower bound is used in place of the unknown ϕ in Theorem 4.2 to yield an upper bound on N. It is not difficult to demonstrate that the upper bound on N obtained using the smallest Markov blanket probability can, at times, be exponentially large, whereas the optimal value of N based on the correct value of $\Pr[x|e]$ is polynomial or even constant. In likelihood weighting, the bound on N for estimating $\Pr[x|e]$ requires a lower bound on the joint probability $\Pr[x, e]$. Shachter and Peot [Shachter and Peot, 1989] use the smallest joint probability in the network for the lower bound on $\Pr[x, e]$. Since the smallest joint probability in an n node binary-valued belief network is at most 2^{-n}, this method requires an exponential number of samples in all cases. Thus, in contrast to the theoretical performance of stochastic simulation algorithms predicted by previous analyses, there has been difficulty in knowing when to stop the simulation in order to guarantee that the output is an (ϵ, δ)-estimate.

Using the Bayesian stopping rule of Section 5.3, we present bounds on the number of samples required that are practical bounds in the sense that they *are* achieved by implementations. With probability $1-\delta$ the bound is optimal for a robust class of prior distributions in the sense that, with probability $1-\delta$, an algorithm stops after the minimum number of simulations. The failure probability δ is made $O(2^{-n})$ by an $O(n)$ multiplicative increase in computational cost. Thus, provided an $O(n)$ increase in the running time is tolerated the probability the algorithm fails to stop within the optimal stopping time is not of significant concern.

The overall running time of stochastic simulation algorithms is factored into the product of the time required to generate samples and the number of samples required to achieve a desired convergence. Based on the Bayesian analysis, Theorem 9.1 gives the number of samples required by a stochastic simulation algorithm to output an (ϵ, δ)-estimate μ of ϕ. In the performance analysis of logic sampling, randomized approximation schemes and likelihood weighting, we first present optimal bounds on the number of samples required to achieve convergence. The bounds we derive assume the prior distribution on the inference under approximation, $\phi = \Pr[x|e]$, conforms to those stated following Theorem 9.1.

For simplicity, in the performance analysis we assume a model of computation where computing a single-node instantiation in logic sampling, computing an unobserved single-node instantiation in likelihood weighting, or computing a transition probability in randomized approximation schemes requires a single unit of computational cost. In an implementation, the true cost of these computations is the product of the cost of running a random number generator and the log of the size of the smallest probability in the model.

8.1 Analysis of Logic Sampling

We analyze separately the running time complexity of both the basic method and the rejection method of logic sampling.

8.1.1 Basic Method

From Section 7.1, the basic method estimates the inference $\Pr[x|e]$ from the estimates of the inferences $\Pr[x, e]$ and $\Pr[e]$. Thus, in the basic method, ϕ is either $\Pr[x, e]$ or $\Pr[e]$. From Corollary 6.2, the number of samples required by the basic method to estimate $\Pr[x, e]$ is, with probability at least $1 - \delta$, bounded by

$$O\left(\frac{1}{\epsilon^2 \Pr[x, e]} \ln \frac{4}{\delta}\right).$$

In other words, the algorithm terminates after this number of samples with probability at least $1-\delta$. At termination, the output μ produced by the basic method is an (ϵ, δ)-estimate.

The time to generate samples in the basic method is equal to the number of nodes that are instantiated in a sample. Thus, we prove Corollary 8.1.

Corollary 8.1 *With probability at least $1 - \delta$, the basic method of logic sampling outputs an (ϵ, δ)-estimate of $\Pr[x|e]$ in running time*

$$O\left(\frac{n}{\epsilon^2 \Pr[x, e]} \ln \frac{4}{\delta}\right).$$

□

8.1.2 Rejection Method

The rejection method estimates $\Pr[x|e]$ directly. From Corollary 6.2, the number of samples required by the rejection method to estimate $\Pr[x|e]$ is,

$$O\left(\frac{1}{\epsilon^2 \Pr[x|e]} \ln \frac{4}{\delta}\right).$$

Once again, the algorithm terminates after this number of samples with probability at least $1 - \delta$, and at termination, outputs an (ϵ, δ)-estimate.

The rejection method, compared to the basic method, employs $O(\frac{1}{\Pr[e]})$ less samples to estimate $\Pr[x|e]$. The decrease in the required number of samples is at the expense of an $O(\frac{1}{\Pr[e]})$ increase in expected running time to generate each sample. We prove Corollary 8.2.

Corollary 8.2 *With probability at least $1-\delta$, the rejection method outputs an (ϵ, δ)-estimate of $\Pr[x|e]$ in running time*

$$O\left(\frac{n}{\epsilon^2 \Pr[x, e]} \ln \frac{4}{\delta}\right).$$

□

Both approaches to logic sampling have similar running times. Conceptually, the difference is whether we spend more time sampling, or more time generating suitable samples. In logic sampling, we arrive at the same running time by either method. In contrast, randomized approximation schemes and likelihood weighting represent variations of the two methods of logic sampling when the choice between a few good samples or many poor samples does affect the running time. Like the rejection method, randomized approximation schemes estimate $\Pr[x|e]$ directly by generating outcomes conditioned on the evidence. And like the rejection method, the time to generate outcomes is expensive. Likelihood weighting subsumes the basic method, and like the basic method, it estimates $\Pr[x|e]$ from estimates of $\Pr[x, e]$ and $\Pr[e]$. The samples are generated efficiently, but the number of samples to achieve convergence is large.

8.2 Analysis of Randomized Approximation Schemes

The Bayesian stopping rule limits the number of runs in multiple short-run algorithms. We cannot apply our methods to straight simulation because it is an example of a single long-run algorithm. We discussed in Section 7.2 that a randomized approximation scheme is a multiple short-runs version of straight simulation. In this section, we apply the Bayesian stopping rule to randomized approximation schemes.

From the discussion in Section 7.3, for randomized approximation schemes ϕ is $\Pr[x|e]$. From Corollary 6.2, a randomized approximation scheme stops after

$$O\left(\frac{1}{\epsilon^2 \Pr[x|e]} \ln \frac{4}{\delta}\right)$$

samples with probability at least $1 - \delta$. At termination, the output μ produced by the randomized approximation scheme is an (ϵ, δ)-estimate.

Like the rejection method of logic sampling, the number of samples used by a randomized approximation scheme is independent of the evidence e—that is, they achieve an $O(\frac{1}{\Pr[e]})$ reduction in the number of required samples compared to the basic method and likelihood weighting. The reduction in the number of samples, however, is at the expense of a significant increase in the time required to generate a sample.

In randomized approximation schemes, the time to generate samples is given by the number of transitions T necessary to satisfy Equation 19. Unfortunately, for complex belief network structures, Dagum and Chavez [Chavez, 1990, Dagum and Chavez, 1991] show that T is exponentially large in the number of nodes of the network.

Corollary 8.3 *With probability at least* $1-\delta$*, a randomized approximation scheme outputs an* (ϵ,δ)*-estimate of* $\Pr[x|e]$ *in running time*

$$O\left(\frac{T}{\epsilon^2 \Pr[x|e]} \ln \frac{4}{\delta}\right),$$

where T *is the time for* ϕ_T *to satisfy Equation 19.* □

8.3 Analysis of Likelihood Weighting

The outcomes of the random variables in likelihood weighting are contained in the interval $[0,a]$ for some $a \geq 1$. Because the outcomes do not form a Bernoulli process, we cannot obtain optimal convergence times for likelihood weighting with the stopping rule of Section 5.4. We use an approximation that gives an upper bound on the number of simulations required to output an (ϵ,δ)-estimate.

From Equation 3, the stopping rule for a $0-a$ random variable is an *upper* bound on the number of samples required for the convergence of a random variable with values in the interval $[0,a]$. How well the upper bound approximates the true number of samples depends on the distribution of the outcomes in the interval $[0,a]$.

Let λ denote the random variable of likelihood weighting

$$\lambda(z) = \chi(z) \cdot \omega(z,e)$$

with with expectation

$$\phi = \Pr[x,e].$$

Let a denote the maximum value of the weighting distribution $\omega(z,e)$. The random variable $\zeta = \frac{1}{a}\lambda$ is $0-1$ valued. We prove the following corollary.

Corollary 8.4 *To output an* (ϵ,δ)*-estimate of* $\Pr[x,e]$*, likelihood weighting requires a number of samples* N *that with probability at least* $1-\delta$ *is*

$$O\left(\frac{a}{\epsilon^2 \Pr[x,e]} \ln \frac{4}{\delta}\right).$$

PROOF The number of samples required when the random variable is $0-a$ valued is an upper bound on the number of samples required when it takes arbitrary values in the interval $[0,1]$. Consider an estimate μ of $\Pr[x|e]$ that is the arithmetic mean of N outcomes λ. It is verified that a sufficient condition for μ to be an (ϵ,δ)-estimate is that N satisfy Theorem 6.1 with estimate $\frac{1}{a}\mu$. Use of Corollary 6.2 completes the proof. □

We cannot *a priori* determine the value of a. Nonetheless, our stopping rule guarantees that the likelihood weighting algorithm stops with very high probability after a number of samples given by Corollary 8.4. To generate a sample, likelihood weighting requires $O(n)$ time to instantiate the unobserved nodes $\mathcal{Z}$. Corollary 8.5 gives the overall running time of likelihood weighting .

Corollary 8.5 *With probability at least* $1-\delta$*, likelihood weighting outputs an* (ϵ,δ)*-estimate of* $\Pr[x|e]$ *in running time*

$$O\left(\frac{an}{\epsilon^2 \Pr[x,e]} \ln \frac{4}{\delta}\right).$$

□

The value of a depends on the specific variant of likelihood weighting used. When the basic algorithm is used, the weighting distribution is the path likelihood. Because the path likelihood is a probability, a must be less than 1. On the other hand, because $\mathbf{E}\omega = \Pr[e]$, there exists some value of z for which $\omega(z,e) \geq \Pr[e]$, and a must be at least $\Pr[e]$. Given the range of a, the number of samples required by the basic algorithm lies intermediate between the number of samples required by logic sampling in Corollary 8.1, $a = 1$, and the number of samples required by a randomized approximation schemes in Corollary 8.3, $a = \Pr[e]$.

The objective of importance sampling is to anticipate the evidential support of e on z, and thus, to adjust the sampling distribution $\tilde{\rho}(z,e)$ until it approximates the posterior distribution $\Pr[z|e]$. The new weighting distribution $\tilde{\omega}(z,e)$ is distributed near $\Pr[e]$; however, when we adjust the sampling distribution we may increase the range of values of $\tilde{\omega}(z,e)$ so the maximum value exceeds 1. Consequently, a is greater than 1 and the algorithm convergences slower than logic sampling. We explain this phenomenon if we recall the definition of the weights in Equation 20. The distribution $\tilde{\rho}(z,e)$ biases the sampling distribution $\rho(z,e)$ and some instances of z are sampled with probability greater than $\rho(z,e)$ while others are sampled with probability less than $\rho(z, e)$. When the sample probability of an instance is reduced, its weight is magnified, and potentially, some weights exceed 1.

9 Summary and Conclusions

Simulation algorithms for probabilistic inference in belief networks operate by using as estimates the fraction of successes of Bernoulli processes. In the limit of infinite trials, the Law of Large Numbers proves these estimates converge to the correct probability. Traditional analyses of the convergence of simulation algorithms for a finite number of trials rely on Zero–One Estimation Theory. This theory gives bounds on number of trials needed to guarantee predefined levels of convergence. Because the bounds are contingent on having *a priori* knowledge of the answer to the initial inferential query, tight bounds on the amount of simulation required cannot be computed.

We developed a Bayesian stopping rule for simulation algorithms that guarantee convergence in a minimum number of trials. We discussed how the Bayesian stopping rules are insensitive to different assumptions about informationless prior probability distributions. We proved that, from a Bayesian perspective, nonBayesian statistical techniques typically discard useful information about inferences of interest. Finally, we applied the stopping rule to analyzing the performance of several popular approximation algorithms for probabilistic inference.

Appendix

Part I: Approximation of Bayesian Error Analysis

We prove Theorem 6.1. Let δ_l and δ_u denote the first and second integrals in Equation 10. Without loss of generality, we assume that $\mu \leq \frac{1}{2}$, and therefore, $\delta_u \geq \delta_l$. Thus δ_u is an upper bound for $\frac{\delta}{2}$.

We approximate the integral expression for δ_u given in Equation 10 by a Reimann summation

$$\delta_u < \sum_{i=1}^{\lfloor \frac{1-\mu}{\epsilon\mu} \rfloor} \epsilon\mu\beta((1+i\epsilon)\mu|\mu,N). \tag{23}$$

We express the beta functions in the summation as

$$\begin{aligned}
\beta((1+i\epsilon)\mu|\mu,N) &= \frac{\mu N(1-\mu)N}{N}\binom{N}{\mu N}[(1+i\epsilon)\mu]^{\mu N-1}[1-(1+i\epsilon)\mu]^{(1-\mu)N-1} \\
&= \frac{\mu N(1-\mu)N}{N}\binom{N}{\mu N}[(1+\frac{1}{2}\epsilon)\mu]^{\mu N-1}[1-(1+\frac{\epsilon}{2})\mu]^{(1-\mu)N-1} \\
&\qquad \times \frac{[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}]^{\mu N}[1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}]^{(1-\mu)N}}{[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}][1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}]} \\
&= \beta((1+\frac{1}{2}\epsilon)\mu|\mu,N)\frac{[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}]^{\mu N}[1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}]^{(1-\mu)N}}{[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}][1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}]}
\end{aligned}$$

Because $\epsilon\mu\beta((1+\frac{1}{2}\epsilon)\mu|\mu,N) \leq 1$ we express Equation 23 as

$$\delta_u < \frac{1}{\left[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right]\left[1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}\right]}\sum_{i=1}^{\lfloor \frac{1-\mu}{\epsilon\mu} \rfloor}\left[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right]^{\mu N}\left[1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}\right]^{(1-\mu)N}$$

Let $a = \frac{1-\mu}{\mu}$. By assumption, $\mu \leq \frac{1}{2}$; hence $a \geq 1$. We express each summand in the upper bound expression for δ_u as

$$\begin{aligned}
\left[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right]^{\mu N}\left[1-\frac{(i-\frac{1}{2})\epsilon\mu}{1-(1+\frac{\epsilon}{2})\mu}\right]^{(1-\mu)N} &= \left[\left(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right)\left(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}\right)^{a}\right]^{\mu N} \\
&= e^{N\mu\ln\left[\left(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right)\left(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}\right)^{a}\right]}.
\end{aligned}$$

Thus,

$$\delta_u \quad < \quad \frac{1}{[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}][1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}]}\sum_{i=1}^{\lfloor \frac{1-\mu}{\epsilon\mu} \rfloor} e^{N\mu\ln\left[(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}})(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}})^{a}\right]}$$

$$< \sum_{i=1}^{\left\lfloor \frac{1-\mu}{\epsilon\mu} \right\rfloor} e^{(N\mu-1)\ln\left[(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}})(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}})^a\right]}.$$

The last inequality follows because

$$\left[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right]\left[1-\frac{(i-\frac{1}{2})\frac{\epsilon}{2}}{1-\frac{\epsilon}{2a}}\right] > \left[1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right]\left[1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}\right]^a.$$

For all $x \geq 0$,

$$\ln(1+x) \leq x,$$

and for all $0 \leq x \leq 1$,

$$\ln(1-x) \leq -x.$$

We now simplify the log-terms in the preceding expression for δ_u.

$$\begin{aligned}
\ln\left[\left(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right)\left(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}\right)^a\right] &= \ln\left(1+\frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}}\right)+a\ln\left(1-\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}}\right) \\
&\leq \frac{(i-\frac{1}{2})\epsilon}{1+\frac{\epsilon}{2}} - a\frac{(i-\frac{1}{2})\frac{\epsilon}{a}}{1-\frac{\epsilon}{2a}} \\
&= -\frac{1}{4}(2i-1)\epsilon^2\left[\frac{1+\frac{1}{a}}{\left(1+\frac{\epsilon}{2}\right)\left(1-\frac{\epsilon}{2a}\right)}\right] \\
&\leq -\frac{1}{6}(2i-1)\epsilon^2
\end{aligned}$$

Thus, we yield

$$\delta_u < \sum_{i=1}^{\left\lfloor \frac{a}{\epsilon} \right\rfloor} e^{-\frac{1}{6}(N\mu-1)\epsilon^2(2i-1)}. \tag{24}$$

We require N to be sufficiently large to make the right hand side of Equation 24 less than 1. Thus, each summand must be less than 1. Therefore,

$$\begin{aligned}
\delta_u &< \sum_{i=1}^{\left\lfloor \frac{a}{\epsilon} \right\rfloor} e^{-\frac{1}{6}(N\mu-1)\epsilon^2(2i-1)} \\
&= e^{-\frac{1}{6}(N\mu-1)\epsilon^2}\left[1+\sum_{i=2}^{\left\lfloor \frac{a}{\epsilon} \right\rfloor-1} e^{-\frac{1}{6}(N\mu-1)\epsilon^2 2(i-1)}\right] \\
&< e^{-\frac{1}{6}(N\mu-1)\epsilon^2}\left[1+\sum_{i=1}^{\left\lfloor \frac{a}{\epsilon} \right\rfloor} e^{-\frac{1}{6}(N\mu-1)\epsilon^2(2i-1)}\right] \\
&< 2e^{-\frac{1}{6}(N\mu-1)\epsilon^2}.
\end{aligned}$$

Using $\frac{\delta}{2} < \delta_u$ and solving for N we get

$$\begin{aligned} N &< \frac{6}{\epsilon^2 \mu} \ln\left(\frac{4}{\delta}\right) + \frac{1}{\mu} \\ &< \frac{7}{\epsilon^2 \mu} \ln\left(\frac{4}{\delta}\right). \end{aligned}$$

Part II: Stopping Rule for Arbitrary Priors

We examine the case of Monte Carlo simulation beginning with an arbitrary prior probability distribution, not necessarily represented as a beta distribution. We develop formulae for computing the distribution of the mean ϕ after we observe N outcomes, given an arbitrary prior probability distribution $\Pr[\phi]$.

Theorem 9.1 *Let ϕ have a prior distribution $\Pr[\phi]$ and let $\pi = \max_{0\leq\phi\leq 1}\Pr[\phi]$. Let k normalize the distribution $\Pr[\phi]\beta(\phi|\mu', N+2)$, where $\mu' = \frac{\mu N}{N+2}$ for estimate μ. For $\epsilon, \delta \leq 1$, define $\hat{\delta} = \frac{\epsilon\mu'\delta}{\pi k^2}$. If N satisfies*

$$N > \frac{4}{\epsilon^2\mu}\ln\left(\frac{4}{\hat{\delta}}\right)$$

then μ is an (ϵ, δ)-estimate.

Proof From Equation 10 and Theorem 5.1 we express the failure probability δ as

$$k\int_0^{\frac{\mu}{(1+\epsilon)}} \Pr[x]\beta(x|\mu', N+2)dx + k\int_{\mu(1+\epsilon)}^1 \Pr[x]\beta(x|\mu', N+2)dx = \delta, \tag{25}$$

where $\mu' = \frac{\mu N}{N+2}$ and

$$k^{-1} = \int_0^1 \Pr[x]\beta(x|\mu', N+2)dx.$$

Let δ_l and δ_u refer to the first and the second integrals in Equation 25. Without loss of generality, we assume that $\mu \leq \frac{1}{2}$, and therefore, $\delta_u \geq \delta_l$. We use the Cauchy-Schwarz inequality for integrals (e.g., see [Purcell, 1972]) to bound δ_u^2 by

$$\delta_u^2 \leq k^2\int_{\mu(1+\epsilon)}^1 \Pr[x]^2 dx \int_{\mu(1+\epsilon)}^1 \beta^2(x|\mu', N+2)dx. \tag{26}$$

By inspection of the proof of Theorem 6.1 given in the Appendix, we verify directly that,

$$\int_{\mu(1+\epsilon)}^1 \beta^2(x|\mu', N+2)dx < \frac{2}{\epsilon\mu'}e^{-\frac{1}{6}(2(N+2)\mu'-2)\epsilon^2}.$$

Furthermore,

$$\begin{aligned}
\int_{\mu(1+\epsilon)}^1 \Pr[x]^2 dx &\leq \max_{\mu(1+\epsilon)\leq x\leq 1}\Pr[x]\int_{\mu(1+\epsilon)}^1 \Pr[x]dx \\
&\leq \max_{0\leq x\leq 1}\Pr[x] \\
&= \pi
\end{aligned}$$

and hence,

$$\delta_u^2 < \frac{2}{\epsilon\mu'}\pi k^2 e^{-\frac{1}{3}((N+2)\mu-1)\epsilon^2}.$$

Let

$$\hat{\delta} = \frac{\epsilon\mu'\delta}{\pi k^2}.$$

Since $\delta < 2\delta_u$ we get,

$$N + 2 > \frac{4}{\epsilon^2\mu'} \ln \frac{4}{\hat{\delta}},$$

or equivalently

$$N > \frac{4}{\epsilon^2\mu} \ln \frac{4}{\hat{\delta}}.$$

□

When $\hat{\delta} \geq \delta^c$ for some constant c, the bound on the number of samples given in Theorem 9.1 is within a constant factor of the bound given in Theorem 6.1. We investigate the conditions which guarantee $\hat{\delta} \geq \delta^c$.

To guarantee efficiency—that is, the number of samples N is within a polynomial factor of $\frac{1}{\mu}$—we always choose the error probability δ to be exponentially small and the error ϵ to be polynomially small in the problem size—for example, 2^{-n} and $\frac{1}{n}$, respectively. For this choice of ϵ and δ, we show that under most circumstances, $\frac{\mu'}{\pi k^2} \geq 2^{-cn}$, and therefore, $\hat{\delta} \geq \delta^{c+2}$.

We assume without loss of generality that there are no deterministic links in the belief network. (When two nodes are connected by a deterministic link, we group the nodes into a single node, and thus, we eliminate the deterministic link.) We assume that all conditional probabilities are greater than 2^{-d} for some constant $d > 1$. (The conditional probabilities of a belief network without deterministic links are rarely less than 10^{-6}.) From Equation 1 it follows that the full joint probability is greater than 2^{-dn}, and therefore, $\mu > 2^{-dn}$. Thus, provided π and k are less than 2^{an} and 2^{bn} for constants a and b, the bound on the number of samples given in Theorem 9.1 is within a constant factor of the bound given in Theorem 6.1. In the extreme case were the prior is concentrated entirely in the interval $[1 - 2^{-an}, 1]$, however, k cannot be bounded by 2^{bn} for any b, even assuming that the prior distribution for μ is bounded by 2^{an}. Nontheless, under most circumstances large variations in the prior do not necessarily lead to large variations in the number of samples required for μ to converge to ϕ.

Acknowledgments

This work was supported by the National Science Foundation under grant IRI-9108385 and by Rockwell International Science Center IR&D funding.

References

[Andreassen et al., 1987] Andreassen, S., Woldbye, M., Falck, B., and Andersen, S. (1987). MUNIN—A causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of 10th International Joint Conference on Artificial Intelligence*, Milan, Italy.

[Beinlich et al., 1989] Beinlich, I., Suermondt, H., Chavez, R., and Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence*, Berlin. Springer-Verlag.

[Billingsley, 1968] Billingsley, P. (1968). *Convergence of Probability Measures*. John Wiley & Sons, New York, NY.

[Breese et al., 1992] Breese, J., Horvitz, E., Peot, M., Gay, R., and Quentin, G. (1992). Automated decision-analytic diagnosis of thermal performance in gas turbines. In *Proceedings of Turbine Expo 92*. American Society of Mechanical Engineers.

[Broder, 1986] Broder, A. (1986). How hard is it to marry at random? (On the approximation of the permanent). In *Proceedings of the Eighteenth ACM Symposium on Theory of Computing*, pages 50–58.

[Chavez, 1990] Chavez, R. (1990). *Architectures and approximation algorithms for probabilistic expert systems.* PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA.

[Chavez and Cooper, 1990] Chavez, R. and Cooper, G. (1990). A randomized approximation algorithm for probabilistic inference on bayesian belief networks. *Networks*, 20:661–685.

[Cooper, 1984] Cooper, G. (1984). *NESTOR: A Computer-based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA. Rep. No. STAN-CS-84-48. Also numbered HPP-84-48.

[Cooper, 1990] Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405.

[Dagum and Chavez, 1991] Dagum, P. and Chavez, R. (1991). Approximating probabilistic inference in Bayesian belief networks. Technical Report KSL-91-46, Knowledge Systems Laboratory, Stanford University, Stanford, CA. To appear in *Pattern Analysis and Machine Intelligence* special issue on Probabilistic Reasoning.

[Dagum and Horvitz, 1991] Dagum, P. and Horvitz, E. (1991). An analysis of Monte-Carlo algorithms for probabilistic inference. Technical Report KSL-91-67, Knowledge Systems Laboratory, Stanford University, Stanford, CA.

[Dagum and Luby, 1991] Dagum, P. and Luby, M. (1991). Approximating probabilistic inference in Bayesian belief networks is NP-hard. Technical Report KSL-91-51, Knowledge Systems Laboratory, Stanford University, Stanford, CA. To appear as a research note in *Artificial Intelligence*.

[DeRobertis and Hartigan, 1981] DeRobertis, L. and Hartigan, J. A. (1981). Bayesian inference using interval measures. *The Annals of Statistics*, 9:235–244.

[Fung and Chang, 1989] Fung, R. and Chang, K. (1989). Weighting and integrating evidence for stochastic simulation in Bayesian networks. In *Proceedings of Fifth Workshop on Uncertainty in Artificial Intelligence, Windsor, ON*, pages 112–117. Association for Uncertainty in Artificial Intelligence, Mountain View, CA.

[Gelman and Rubin, 1991] Gelman, A. and Rubin, D. B. (1991). An overview and approach to inference from iterative simulation. In *Workshop on Bayesian Computation via Stochastic Simulation*.

[Gradshteyn and Ryzhik, 1980] Gradshteyn, I. S. and Ryzhik, I. M. (1980). *Table of Integrals, Series, and Products*. Academic Press, Inc., New York, NY.

[Hartigan, 1983] Hartigan, J. A. (1983). *Bayes Theory*. Springer-Verlag, New York, NY.

[Heckerman et al., 1992] Heckerman, D., Horvitz, E., and Nathwani, B. (1992). Toward normative expert systems: The Pathfinder project. *Method of Information in Medicine*.

[Henrion, 1988] Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Uncertainty in Artificial Intelligence* **2**, pages 149–163. North-Holland, Amsterdam.

[Henrion, 1990] Henrion, M. (1990). Towards efficient probabilistic diagnosis in multiply connected networks. In Oliver, R. and Smith, J., editors, *Influence Diagrams, Belief Networks, and Decision Analysis*, pages 385–409. John Wiley and Sons, Chichester.

[Henrion et al., 1992] Henrion, M., Breese, J., and Horvitz, E. (1992). Decision analysis and expert systems. *AI Magazine*, 12:64–91.

[Horvitz et al., 1988] Horvitz, E., Breese, J., and Henrion, M. (1988). Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2:247–302.

[Horvitz et al., 1989] Horvitz, E., Suermondt, H., and Cooper, G. (1989). Bounded conditioning: Flexible inference for decisions under scarce resources. In *Proceedings of Fifth Workshop on Uncertainty in Artificial Intelligence, Windsor, ON*, pages 182–193. Association for Uncertainty in Artificial Intelligence, Mountain View, CA.

[Howard, 1970] Howard, R. (1970). Decision analysis: Perspectives on inference, decision, and experimentation. *Proceedings of the IEEE*, 58:632–643.

[Jerrum et al., 1986] Jerrum, M., Valiant, L., and Vazirani, V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188.

[Karp et al., 1989] Karp, R., Luby, M., and Madras, N. (1989). Monte-Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10:429–448.

[Lauritzen and Spiegelhalter, 1988] Lauritzen, S. and Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society* **B**, 50(19).

[Miller et al., 1986] Miller, R., McNeil, M., Challinor, S., Masarie, F., and Myers, J. (1986). The INTERNIST-1/Quick Medical Reference project–status report. *Western Journal of Medicine*, 145:816–822.

[Nino-Murcia and Shwe, 1991] Nino-Murcia, G. and Shwe, M. (1991). An expert system for diagnosis of sleep disorders. In *1991 Annual Meeting Abstracts*, page 165, Toronto, CA. Association of Professional Sleep Societies.

[Pearl, 1987a] Pearl, J. (1987a). Addendum: Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 33:131.

[Pearl, 1987b] Pearl, J. (1987b). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32:245–257.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann, San Mateo, CA.

[Purcell, 1972] Purcell, E. J. (1972). *Calculus with Analytic Geometry, 2nd edition.* Appleton-Century-Crofts, New York.

[Raftery and Lewis, 1992] Raftery, A. E. and Lewis, S. (1992). How many iterations in the gibbs sampler. In *Bayesian Statistics 4*, pages 765–776. Oxford University Press, Oxford.

[Rubinstein, 1981] Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method.* John Wiley & Sons, New York, NY.

[Rutledge et al., 1989] Rutledge, G., Thomsen, G., Beinlich, I., Farr, B., Sheiner, B., and Fagan, L. (1989). Combining qualitative and quantitative computation in a ventilator therapy planner. In *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care*, pages 315–319, Washington, D.C. The American Medical Informatics Association.

[Seneta, 1973] Seneta, E. (1973). *Non-negative matrices and Markov chains, 2nd ed.* Springer-Verlag, New York, NY.

[Shachter and Peot, 1989] Shachter, R. and Peot, M. (1989). Evidential reasoning using likelihood weighting. *Artificial Intelligence*. Forthcoming.

[Shachter and Peot, 1990] Shachter, R. and Peot, M. (1990). Simulation approaches to general probabilistic inference on belief networks. In *Uncertainty in Artificial Intelligence* **5**, pages 221–231. Elsevier, Amsterdam.

[Shwe and Cooper, 1991] Shwe, M. and Cooper, G. (1991). An empirical analysis of likelihood-weighting simulation on a large, multiply connected medical belief network. *Computers and Biomedical Research*, 24:453–475.

[Shwe et al., 1991a] Shwe, M., Middleton, B., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H., and Cooper, G. (1991a). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base—I: The probabilistic model and inference algorithms. *Methods of information in medicine*, 30:241–255.

[Shwe et al., 1991b] Shwe, M., Middleton, B., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H., and Cooper, G. (1991b). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base–II: Evaluation of diagnostic performance. *Methods of information in medicine*, 30:256–267.

[Sinclair and Jerrum, 1989] Sinclair, A. and Jerrum, M. (1989). Approximate counting, uniform generation and rapidly mixing Markov chains. *Inform. Comput.*, 82:93–133.

[von Neumann, 1951] von Neumann, J. (1951). *Nat. Bureau Standard Appl. Math. Series*, 12:36–39.