Approximating Probabilistic Inference in Bayesian Belief Networks

Paul Dagum and R. Martin Chavez

Abstract— **A belief network comprises a graphical representation of dependencies between variables of a domain and a set of conditional probabilities associated with each dependency. Unless *P=NP*, an efficient, exact algorithm does not exist to compute probabilistic inference in belief networks. Stochastic simulation methods, which often improve run times, provide an alternative to exact inference algorithms. We present such a stochastic simulation algorithm $\mathcal{D}$-BNRAS that is a randomized approximation scheme. To analyze the run time, we parameterize belief networks by the *dependence value* $\mathcal{D}_\xi$, which is a measure of the cumulative strengths of the belief network dependencies given background evidence ξ. This parameterization defines the class of *f-dependence networks*. The run time of $\mathcal{D}$-BNRAS is polynomial when f is a polynomial function. Thus, the results of this paper prove the existence of a class of belief networks for which inference approximation is polynomial and, hence, provably faster than *any* exact algorithm.**

I. Introduction

BELIEF NETWORKS denote a knowledge representation that is ideally suited to model uncertainty in complex domains. Belief networks are the paradigm of knowledge representation in medical decision systems. The intractability of probabilistic inference in large belief networks, however, impedes their application to large domains. Cooper [6] proves probabilistic inference for belief networks is NP-hard. Consequently, we do not expect general-purpose algorithms for probabilistic inference to run in polynomial time. The need to solve time-pressured decision problems in medical applications motivates researchers to design approximation algorithms that trade complexity in run time for accuracy of computation. Stochastic simulation algorithms such as forward propagation [10], [11], [18], [19] and Gibbs sampling [1], [3], [4], [15], [16] number among such algorithms.

For many classes of inputs, stochastic simulation algorithms for probabilistic inference require exponential run time [3], [4], [16]. For example, logic sampling [11] and likelihood weighting [18] require exponential run time on inferences conditioned on rare observations. More generally, Dagum and Luby [9] prove that the approximation of probabilistic inference is NP-hard. Thus, they confirm that all stochastic simulation algorithms exhibit poor behavior on certain classes of belief networks. In spite of this negative result, the plethora of belief network applications in medical expert systems compels us to search for approximation algorithms that run faster than exact algorithms, even though we know that the approximation algorithms do not run in polynomial time on certain classes of belief-network inputs.

Computer scientists formulate stochastic simulation algorithms as *randomized approximation schemes* (RAS's) [14]. A stochastic simulation algorithm for probabilistic inference is a RAS if on inputs $\epsilon, \delta \leq 1$ and inference $\Pr[X = x|\xi]$, the output lies within relative error ϵ of $\Pr[X = x|\xi]$ with probability of at least $1 - \delta$. Given the parameters of the approximation, ϵ and δ, a RAS provides *a priori* bounds on the required run time. A RAS for probabilistic inference has desirable properties. For example, automated medical support systems regularly face time-pressured decision problems. The *a priori* bound of a RAS allows resource constraints to determine the accuracy of the approximation. Since a RAS incrementally tightens the error bounds, the system may make a treatment recommendation immediately or rather defer a recommendation and continue to reason. A rational decision results from a utility model weighing the expected value of further computation against the cost of inference-based delay.

Central to the formulation of a RAS, the zero-one estimator theorem bounds the number of belief network instantiations N required by a RAS to output an estimate of the input inference [14]. For input inference $\Pr[X = x|\xi]$, this number is proportional to $\Pr[X = x|\xi]^{-1}$. Thus, evaluation of N given by the zero-one estimator theorem requires prior knowledge of $\Pr[X = x|\xi]$. Because this prior knowledge is unknown in advance, BNRAS, likelihood weighting, and logic sampling algorithms employ easily computable lower bounds on the inference to yield upper bounds on N. Unfortunately, this approach often is conservative; it yields an upper bound on the number of simulations N that exceeds the optimal value provided by the zero-one estimator theorem by an exponential factor—for example, a factor of 2^n on an n-node belief network. Furthermore, it is NP-hard to determine whether N is finite. By the zero-one estimator theorem, N is finite if and only if $\Pr[X = x|\xi]$ is nonzero: an NP-hard decision problem [6]. Inferences close to zero present a further problem to stochastic simulation algorithms that employ the zero-one estimator theorem. Approximations based naively on zero-one estimation theory are intractable even for exact evaluations of N.

Manuscript received November 1991; revised August 1992. This work was supported by National Library of Medicine Grants LM-04 136, LM-05 208, National Science Foundation Grant IRI-9 108 385, and Rockwell International Science Center IR&D funds.

P. Dagum is with the Section on Medical Informatics, Stanford University School of Medicine, Stanford, CA 94305-5479. He is also with Rockwell Palo Alto Laboratory, Palo Alto, CA.

R. M. Chavez is with the Section on Medical Informatics, Stanford University School of Medicine, Stanford, CA 94305-5479.

IEEE Log Number 9206610.

The algorithm BNRAS of Chavez and Cooper [4] represents the first design of a stochastic simulation algorithm for probabilistic inference formulated explicitly as a RAS. The formulation of the algorithm utilizes Markov simulation. Nevertheless, it does not exploit the geometric properties of the Markov chain. Thus, Chavez and Cooper obtain suboptimal results on the convergence of the simulation.

We characterize belief networks by their *dependence value* $\mathcal{D}$. Intuitively, the dependence value measures the cumulative strength of the dependencies among nodes in a belief network encoded in the conditional probabilities of each node. We present an approximation algorithm $\mathcal{D}$-BNRAS for probabilistic inference in the spirit of Chavez and Cooper's BNRAS [4], [6]. However, we focus on the geometric intuition underlying the algorithm: a random walk on a hypercube. Each hypercube vertex represents a possible instantiation of the network nodes, and each hypercube edge connects vertices if they differ in the instantiation of a single node. We exploit this geometry to prove tight convergence bounds on the random walk. Consequently, we derive nonasymptotic results on the rate of convergence by proving a lower bound on the conductance of the hypercube with results from [20]. We use these bounds to reduce the time required by the algorithm to output an instantiation. A rigorous analysis shows that $\mathcal{D}$-BNRAS significantly improves the run time of BNRAS on all classes of inputs.

The class of *f-dependence networks* comprises belief networks with a dependence value bounded by the function $f(n)$. We prove $\mathcal{D}$-BNRAS runs in time proportional to $f^4(n)$. When $f(n)$ is a polynomial function, $\mathcal{D}$-BNRAS has polynomial run time. If for all n, $f(n) \geq 1 + \alpha$ for any constant $\alpha > 0$, then probabilistic inference is NP-hard for f-dependence networks. Thus, we do not expect to find efficient exact algorithms for probabilistic inference even for very restricted f-dependence networks. $\mathcal{D}$-BNRAS runs in polynomial time when f is a polynomial, and it yields a tractable solution to the problem of probabilistic inference for this class.

Formulation of stochastic simulation algorithms for probabilistic inference as RAS algorithms involves the naive application of the zero-one estimator theorem. We encounter two difficulties with this approach: We must compute good lower bounds on the inference we intend to approximate, and when the inference nears zero, we must use a very large number of simulations. In contrast with the difficulties encountered by these algorithms, we prove a key result that allows $\mathcal{D}$-BNRAS to employ a polynomial number of instantiations to approximate *any* input inference. Thus, the efficiency of $\mathcal{D}$-BNRAS is independent of the input inference; however, the efficiency relies strongly on the dependence value of the belief network. Thus, in almost all cases, $\mathcal{D}$-BNRAS requires fewer instantiations to output an estimate than previous stochastic simulation algorithms, but the run time to generate instantiations is significantly longer than likelihood weighting and, in many cases, logic sampling.

II. Background

Here and elsewhere, B denotes a binary-valued belief network on n *unobserved* nodes $\{X_1, \ldots, X_n\}$ with background evidence ξ. The set X refers to an arbitrary subset of unobserved nodes in B. For any node X_i, any set of unobserved parents $\boldsymbol{u}_{X_i}$, and any set of observed parents ξ_{X_i}, a belief network specifies a conditional probability function $\Pr[X_i|\boldsymbol{u}_{X_i}, \xi_{X_i}]$. We simplify the notation of the conditional probability function by writing $\Pr[X_i|\boldsymbol{u}_{X_i}, \xi]$, where it is understood that ξ, in this context, refers to ξ_{X_i}. The full joint probability distribution specified by a belief network is calculated by taking the product of the conditional probabilities

$$\Pr[X_1, \ldots, X_n, \xi] = \prod_{i=1}^{n} \Pr[X_i|\boldsymbol{u}_{X_i}, \xi]. \tag{1}$$

Probabilistic inference in belief networks refers to the computation of $\Pr[X = x|\xi]$ for an instantiation x of X and background evidence ξ.

A. Dependence Value

We parametrize belief networks by their *dependence value* $\mathcal{D}_\xi \geq 1$. The dependence value of a belief network depends on the background evidence ξ. Intuitively, the dependence value gives a measure of the cumulative strength of the dependencies among nodes in a belief network that are encoded by the conditional probabilities associated with each node.

For each node X_i, we define l_i and u_i as the greatest and smallest numbers, respectively, such that, for instantiation x_i of X_i, and for all instantiations of X_i's unobserved parents $\boldsymbol{u}_{X_i}$

$$l_i \leq \Pr[x_i|\boldsymbol{u}_{X_i}, \xi] \leq u_i.$$

It follows that

$$(1 - u_i) \leq \Pr[\overline{x}_i|\boldsymbol{u}_{X_i}, \xi] \leq (1 - l_i)$$

where $\overline{x}_i = 1 - x_i$. Note that $l_i > 0$ and $u_i < 1$ since we assume that no complete instantiation of the network has zero probability. Let $\lambda_i = \max\left(\frac{u_i}{l_i}, \frac{1-l_i}{1-u_i}\right)$. When X_i is a prior node—that is, X_i has no parents—or when $\boldsymbol{u}_{X_i}$ is empty—that is, the parents of X_i have been observed—then $\lambda_i = 1$.

Definition: For a belief network B, the *dependence value* is given by

$$\mathcal{D}_\xi = \prod_{i=1}^{n} \lambda_i.$$

By this definition, $\mathcal{D}_\xi \geq 1$. The trivial case when $\mathcal{D}_\xi = 1$ occurs when there are no conditional dependencies between nodes in the belief network.

Definition: Let $f(n)$ denote a positive-valued function on the positive integers. The class of *f-dependence networks* consists of the set of belief network and evidence couples (B, ξ) such that if B has n nodes and dependence value $\mathcal{D}_\xi$, then $\mathcal{D}_\xi \leq f(n)$.

When $f(n)$ is a polynomial function, the class of f-dependence networks is an example of a class of *polynomial dependence networks*.

B. Randomized Approximation Schemes

Convergence analysis of simulation algorithms in the theoretical computer science community is rooted in zero-one estimation theory. The methodology carries over to the analysis of simulation algorithms for probabilistic inference [18], [4], [6].

We review briefly the zero-one estimation approach for convergence anlaysis of a simulation algorithm. Consider the problem of trying to estimate $\Pr[\mathcal{A}]$ in the probability space $(\Omega, 2^{\Omega}, \Pr)$, where $\mathcal{A}$ denotes a subset of Ω. Define the random variable $\zeta = \zeta(\omega)$ to take on the value 1 when $\omega \in \mathcal{A}$ and to take on the value 0 otherwise. The Monte Carlo method simulates the probability Pr and scores the random variable $\zeta = \zeta(\omega)$ to estimate $\phi = \Pr[\mathcal{A}]$. By the Law of Large Numbers, in the limit of an infinite number of trials, the arithmetic mean μ of the output of each simulation converges to ϕ. After a finite number of trials N, the current fraction μ estimates ϕ.

For simulation algorithms, we desire an upper bound on N that guarantees that μ provides 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

$$\Pr[\phi(1+\epsilon)^{-1} \leq \mu \leq \phi(1+\epsilon)] > 1 - \delta. \tag{2}$$

A RAS for probabilistic inference is a randomized algorithm that accepts as input a belief network B, instantiation $X = x$, and two positive parameters ϵ and δ. The output of the algorithm is an estimate μ of ϕ that satisfies (2).

The zero-one estimator theorem gives the smallest number of trials required for a RAS to satisfy (2)

$$N = \frac{4}{\phi\epsilon^2} \log \frac{2}{\delta}. \tag{3}$$

For details on this derivation, see [14]. When N, ϵ, and δ satisfy (3) then μ satisfies (2).

The upper bound on N provided by the zero-one estimator theorem is contingent on ϕ—the same quantity we estimate with a simulation algorithm to estimate. To circumvent this circular definition, polynomial time computable lower bounds for ϕ allow us to derive an upper bound estimate of N. A key challenge is the computation of a lower bound within a constant multiplicative factor of ϕ to avoid wasteful computations. Unfortunately, in many cases, the best computable lower bound is $O(2^{-n})$.

C. Stochastic Simulation Algorithms for Probabilistic Inference

Stochastic simulation algorithms for probabilistic inference include logic sampling [11], straight simulation [15], [16], the randomized approximation scheme BNRAS [4], [5] and likelihood weighting [18] algorithms. Chavez and Cooper [4], [3] reformulate straight simulation and logic sampling as RAS algorithms. The number of simulations necessary for convergence is determined by the zero-one estimator theorem. Furthermore, Chavez and Cooper present the first explicitly designed RAS: BNRAS [3]. Thus, Chavez and Cooper show that (3) bounds the number of trials required by these algorithms to output estimates that satisfy (2). Similarly, Shachter and Peot [18] reformulate likelihood weighting as a RAS and analyze convergence using the zero-one estimator theorem. We now discuss two limitations of stochastic-simulation algorithms relying on (3) to bound the number of trials.

We denote the inference probability $\Pr[X = x|\xi]$ by ϕ. We established that we require ϕ to determine N. Since ϕ is the probabilistic inference the stochastic simulation algorithm computes, it is not known in advance. We resolve the problem with a computable lower bound on ϕ that provides an upper bound on N. In the absence of a good lower bound on ϕ, many wasteful trials are generated to estimate ϕ. For example, Shachter and Peot [18] use the smallest inference probability in the network to bound ϕ. More generally, it is NP-hard to determine whether $\phi > 0$ [6], and thus, by (3), it is NP-hard to determine whether N is finite.

When inferences approach zero, stochastic simulation algorithms that employ (3) to bound the number of trials experience a further complication. From this equation, the number of trials becomes intractable because N approaches infinity as ϕ approaches zero. When ϕ is an inference conditioned on evidence—for example, $\phi = \Pr[X = x|\xi]$—then logic sampling and likelihood weighting algorithms are subject to this complication. These algorithms do not estimate ϕ directly. Rather, they estimate $\Pr[X = x, \xi]$ and $\Pr[\xi]$. Bayes' rule dictates that the ratio of these two estimates is an estimate of ϕ. The number of trials required to estimate $\Pr[X = x, \xi]$, or to estimate $\Pr[\xi]$, is given by (3), with ϕ denoting $\Pr[X = x, \xi]$ or $\Pr[\xi]$. When the evidence ξ is rare or when it contains many observed nodes, the joint probability $\Pr[X = x, \xi]$ is small, and logic sampling or likelihood weighting algorithms require many trials to output an estimate of $\Pr[X = x|\xi]$.

In contrast, the run time of $\mathcal{D}$-BNRAS is independent of ϕ. Furthermore, unlike logic sampling and likelihood weighting, $\mathcal{D}$-BNRAS performs better when the evidence set is large. The effect of multiple observations reduces the dependence value $\mathcal{D}_{\xi}$ of the network and, therefore, speeds the generation of trials. $\mathcal{D}_{\xi}$-BNRAS, however, requires a long run time to generate a simulation, whereas logic sampling and likelihood weighting generate simulations efficiently.

III. The Algorithm $\mathcal{D}$-BNRAS

Given a belief network and evidence set, we construct a Markov process. If we simulate this process for sufficient time, then we sample the joint probability distribution conditioned on the evidence. We prove results on the convergence of the Markov process and, hence, on the simulation time required to sample this distribution. We use the the trial generator to construct $\mathcal{D}$-BNRAS. We then prove results on the number of samples $\mathcal{D}$-BNRAS requires to achieve a specified precision in the estimate. We combine this result with the time to generate a sample, and thus, we obtain the run time of $\mathcal{D}$-BNRAS.

A. The Trial Generator

We construct a trial generator for belief networks—that is, an ergodic Markov process $\mathcal{MC}$ on the space of instantiations of B. The joint probability distribution conditioned on the evidence ξ, $\Pr[\ |\xi]$ represents the stationary distribution of

the time-reversible ergodic Markov chain $\mathcal{MC}$. Thus, in its stationary distribution, $\mathcal{MC}$ samples $\Pr[\ |\xi]$. The process $\mathcal{MC}$ reaches the stationary distribution only in the limit of an infinite number of simulations; for finite simulation, $\mathcal{MC}$ approximates the stationary distribution. In Section III-E, we will incorporate the error from $\mathcal{MC}$ into the error ϵ of (2).

Without loss of generality, we restrict the presentation to belief networks with binary valued nodes and with nonzero conditional probabilities. For $0 \leq i \leq 2^n$, let i denote both the instantiation of the nodes in B to the binary representation of i and, in addition, a binary representation of a node in an n-dimensional hypercube (n-cube). Let $e_1, \ldots, e_n$ denote the basis of the n-cube, where e_l is the vector with coordinate l set to 1 and all other coordinates set to 0. Let $\oplus$ denote the symmetric-difference operator. B defines the Markov chain $\mathcal{MC}$ as follows:

1) With probability $\frac{1}{2}$, from any state i, randomly choose an l such that $1 \leq l \leq n$, and make a transition to state $j = i \oplus e_l$ with probability

$$\frac{\Pr[j|\xi]}{\Pr[i|\xi] + \Pr[j|\xi]}.$$

2) With probability $\frac{1}{2}$, do nothing—that is, make a null transition to the same state.

Thus, the transition probabilities of the Markov chain from i to the neighbor $j = i \oplus e_l$ are given as follows:

$$P_{ij} = \begin{cases} 1 - \sum_{k \neq i} P_{ik}, & i = j; \\ \frac{1}{2n} \frac{\Pr[j|\xi]}{\Pr[i|\xi] + \Pr[j|\xi]}, & i \neq j. \end{cases}$$

Note that the self-loop probability P_{ii} is defined to normalize the probability of making a transition.

By the definition of an n-cube and from the equivalence between instantiations of B and nodes in the n-cube, it follows that the Markov chain $\mathcal{MC}$ is a random walk on the n-cube. The self-loop probability renders the chain aperiodic; the existence of a path from any state to every other state makes the chain irreducible; therefore, the chain is ergodic. Symmetry considerations dictate that the chain is time reversible.

The ergodicity of the chain guarantees a unique stationary distribution. We show the stationary distribution is identical to the belief network's joint-probability distribution Pr.

Lemma 1: The stationary distribution of the Markov chain $\mathcal{MC}$ is the joint probability distribution Pr of B.

Proof: From the theory of ergodic Markov chains, it suffices to show, for any state i, Pr satisfies the eigenvalue equation

$$\Pr[i|\xi] = \sum_j \Pr[j|\xi] P_{ji}.$$

However

$$\begin{aligned} \sum_j \Pr[j|\xi] P_{ji} &= \frac{1}{2n} \sum_{j \neq i} \Pr[j|\xi] \frac{\Pr[i|\xi]}{\Pr[i|\xi] + \Pr[j|\xi]} + \Pr[i|\xi] P_{ii} \\ &= \frac{1}{2n} \sum_{j \neq i} \Pr[i|\xi] \frac{\Pr[j|\xi]}{\Pr[i|\xi] + \Pr[j|\xi]} + \Pr[i|\xi] P_{ii} \\ &= \Pr[i|\xi] \left(\sum_{j \neq i} \frac{1}{2n} \frac{\Pr[j|\xi]}{\Pr[i|\xi] + \Pr[j|\xi]} + P_{ii} \right) \\ &= \Pr[i|\xi] \sum_j P_{ij} \\ &= \Pr[i|\xi], \end{aligned}$$

which proves the lemma. □

B. The Approximation Algorithm $\mathcal{D}$-BNRAS

A recurring theme in stochastic simulation algorithms for probabilistic inference is the slow convergence of algorithms when the computed inferences are small. The problem traces back to the result of the zero-one estimator theorem in (3). We described, in Section II-C, how stochastic simulation algorithms traditionally perform poorly on inferences near zero and, furthermore, how users of these algorithms experience the problem of obtaining reasonable lower bounds on the inference probabilities. A poor lower bound translates, through (3), to a large value for N and, therefore, an excessive number of trials to approximate an inference. Dagum and Horvitz offer the most general solution to the latter problem [8]. They develop optimal Bayesian stopping rules for stochastic simulation algorithms. However, we need to address the intractability problem encountered by previous simulation algorithms when inferences are near zero.

$\mathcal{D}$-BNRAS solves the preceding problems very efficiently with the self reducibility of probabilistic inference. $\mathcal{D}$-BNRAS decomposes the problem of estimating an inference probability into one of estimating inferences for a set of subproblems. The decomposition guarantees that the subproblem inference probabilities are at least one half. Thus, by (2), each subproblem inference can be approximated with N trials, where

$$N = \frac{16}{\epsilon^2} \log \frac{2}{\delta}. \tag{4}$$

In the subsequent analysis, we simplify the presentation if we assume the trial generator outputs instantiations of B with probability distribution $\Pr[\ |\xi]$. In reality, however, the distribution of the trial generator $\mathcal{MC}$ only approximates this distribution for finite simulation runs. In Section III-E, we show how the approximation can be made sufficiently close to $\Pr[\ |\xi]$ to render our results valid.

We introduce the algorithm $\mathcal{D}$-BNRAS by considering the problem of estimating the inference $\phi = \Pr[X = x|\xi]$. For $k = 1, \ldots, n$, let ϕ_k denote the probabilities

$$\phi_k = \Pr[X = x_1, \ldots, X = x_k, \xi]$$

and let

$$\phi_0 = \Pr[\xi].$$

Without loss of generality, we assume the set of nodes X contains nodes $X_1, \ldots, X_p$ instantiated to $x_1, \ldots, x_p$. Thus, we express the inference ϕ as

$$\phi = \frac{\phi_p}{\phi_0}. \tag{5}$$

$\mathcal{D}$-BNRAS outputs estimates of the inferences ϕ_p and ϕ_0. The ratio of estimates in (5) provides an estimate of ϕ.

Using Bayes' theorem

$$\phi_p \Pr[x_{p+1}|x_1, \ldots, x_p, \xi] = \phi_{p+1}. \tag{6}$$

Equation (6) reduces the problem of estimating ϕ_p to the problems of 1) estimating ϕ_{p+1} and 2) of estimating $\Pr[x_{p+1}|x_1, \ldots, x_p, \xi]$. Estimating ϕ_{p+1} is not visibly simpler than estimating ϕ_p. However, solving problem (2) is a simpler problem when the instantiation x_{p+1} is chosen appropriately. Observe

$$\Pr[x_{p+1}|x_1, \ldots, x_p, \xi] + \Pr[\overline{x}_{p+1}|x_1, \ldots, x_p, \xi] = 1 \tag{7}$$

where, in general, $\overline{x}_i$ denotes the instantiation $1 - x_i$. We briefly discuss how to choose the instantiation x_{p+1} such that the first probability in (7) exceeds one half. The estimation of $\Pr[x_{p+1}|x_1, \ldots, x_p, \xi]$ is the first of the easily solvable subproblems that $\mathcal{D}$-BNRAS generates.

Similarly, we reduce the estimation of ϕ_{p+1} to 1) the estimation of ϕ_{p+2} and to 2) the easily solvable subproblem $\Pr[x_{p+2}|x_1, \ldots, x_{p+1}, \xi]$ when x_{p+2} is chosen so that this probability has a value of at least one half. We continue to generate easily solvable subproblems until we reach ϕ_n. At this point, we have reduced the estimation of ϕ_p to a set of n-p easily solvable subproblems and the estimation of ϕ_n. However, ϕ_n is the probability of a complete instantiation of the nodes in B, and we compute it exactly with 1.

In summary, we yield an estimate of ϕ_p after $\mathcal{D}$-BNRAS completes *n-p cycles*. In cycle i, the algorithm runs N trials, and it separately scores μ and $\overline{\mu}$, which are the estimates of $\Pr[x_{p+i}|x_1, \ldots, x_{p+i-1}, \xi]$ and $\Pr[\overline{x}_{p+i}|x_1, \ldots, x_{p+i-1}, \xi]$, respectively. Therefore, one of μ or $\overline{\mu}$ exceeds one half. If we run the algorithm for a number of trials N

$$N = 16\frac{(n-p)^2}{\epsilon^2}\log\frac{2}{\delta}, \tag{8}$$

then one of μ or $\overline{\mu}$ satisfies (2) with error $\epsilon(n-p)^{-1}$. Without loss of generality, we assume the estimate μ satisfies (2) with error $\epsilon(n-p)^{-1}$. We instantiate node X_{p+1} to x_{p+i}, and we store μ.

At the end of the n-p cycles, we compute the probability ϕ_n using (10). We divide ϕ_n by the product of the μ's stored at the end of each cycle to estimate ϕ_p. The estimates of each subproblem satisfy (1) with error $\epsilon(n-p)^{-1}$. Thus, standard error propagation dictates that the estimate of ϕ_p satisfies this equation with error ϵ.

C. Convergence Bounds for $\mathcal{MC}$

The *relative pointwise distance* (RPD) measures the distance between the stationary distribution of $\mathcal{MC}$ and the distribution after simulation of $\mathcal{MC}$ for T transitions. Thus, the RPD gives a measure, parametrized by the number of transitions T, of the error in the trial-generation phase occuring because we sample the stationary distribution after a finite number of transitions of a Markov process. For a given error tolerance, we chose T so that the RPD is less than the given error. However, we cannot compute the RPD directly. Instead, we use Theorem 2 to relate the RPD to a computable graph-theoretic property of a Markov process known as the *conductance*.

We begin with some observations about $\mathcal{MC}$. The Markov process $\mathcal{MC}$ is *time reversible* since, for any pair of states $i, j, \Pr[i|\xi]P_{ij} = \Pr[j|\xi]P_{ji}$. For a time-reversible Markov chain, the *underlying graph* H of the chain is the weighted undirected graph with vertex set $[m]$, and for any $i, j \in [m]$, edge (i, j) has weight $w_{ij} = \Pr[i|\xi]P_{ij} = \Pr[j|\xi]P_{ji}$. (If $P_{ij} = 0$, then edge (i, j) is not in the graph.) Jerrum and Sinclair define the conductance of the graph H as

$$\Phi(H) = \min_S \left\{ \frac{\sum_{i \in S, j \notin S} \Pr[i|\xi]P_{ij}}{\sum_{i \in S} \Pr[i|\xi]} \right\} \tag{9}$$

with minimization performed over all subsets S of $[m]$ [12].

Let $P_{ij}^{(T)}$ denote the T-step transition probability from state i to j—that is, the probability we reach state j after T transitions if we start in state i. Define the RPD

$$\Delta(T) = \max_{i,j \in [m]} \left\{ \frac{\left| P_{ij}^{(T)} - \Pr[j|\xi] \right|}{\Pr[j|\xi]} \right\}. \tag{10}$$

By relating the conductance to the second eigenvalue of the chain's transition matrix, which governs the transient behavior of the chain, Jerrum and Sinclair prove the following upper bound on the RPD after t transitions [12].

Theorem 2: Let $H = (V(H), E(H))$ represent the underlying graph of the time-reversible Markov chain $\mathcal{MC}$ with conductance $\Phi(H)$, stationary distribution $\Pr[\,|E]$, and minimum self-loop probability $\frac{1}{2}$. Let $\Pi = \min_{i \in [m]} \Pr[i|\xi]$. Then, the RPD is bounded by

$$\Delta(T) \leq \frac{\left(1 - \Phi(H)^2/2\right)^T}{\Pi}.$$

□

Theorem 2 provides an upper bound on the number of transitions required for the RPD of $\mathcal{MC}$ to be within a given tolerance ζ:

$$T \leq 4 \cdot \frac{\log \frac{1}{\zeta\Pi}}{\Phi(H)^2}. \tag{11}$$

D. Conductance of $\mathcal{MC}$

In this section, we prove a lower bound on the conductance of the underlying graph of $\mathcal{MC}$. A lower bound on the conductance, in conjunction with Theorem 2, gives us an upper bound on the RPD of $\mathcal{MC}$ after T transitions. Thus, we choose T so that the probability distribution of the instances generated by $\mathcal{MC}$ lies within a given error tolerance ζ of the correct, or stationary, distribution.

The dependence value of a belief network parameterizes the lower bound on the conductance. For polynomial dependence

networks, the conductance is sufficiently large to guarantee rapid convergence.

Theorem 3: For belief network B with dependence value $\mathcal{D}_\xi$, the underlying graph H of the Markov chain $\mathcal{MC}$ has conductance satisfying

$$\Phi(H) \geq \Phi_l(H) = \frac{p_0}{2\mathcal{D}_\xi^2}$$

where p_0 denotes the minimum transition probability of $\mathcal{MC}$.

Proof: We use the methods developed by Jerrum and Sinclair [12]. The graph H is an n-cube with weighted edges on vertex set $[n]$ consisting of 2^n binary vectors of length n. Each vertex in $[n]$ is identified with an instantiation of B in the usual way. Let $\Pr[\,|\xi]$ denote the stationary distribution of $\mathcal{MC}$. For any $S \subset [n]$, where $[n]$ denotes the vertex set of H, let $\overline{S} = [n] \setminus S$, and let $C(S)$ represent the edge cutset of S—that is, $(i,j) \in C(S)$ if and only if vertices i and j are neighbors in the hypercube such that $i \in S$ and $j \in \overline{S}$. Define *unique paths* in H between any two states as follows. Let $i = (x_{i1} \cdots x_{in}) \in S$ and $j = (x_{j1} \cdots x_{jn}) \in \overline{S}$. The path from i to j is the following. Examine the first coordinate x_{i1} in the binary vector i. If it has the same value as the first coordinate x_{j1} in j, proceed to the next coordinate; otherwise, move from i to the neighboring state $(x_{j1}x_{i2} \cdots x_{in})$. By sequentially progressing through all the coordinates, this procedure defines a unique path from i to j.

The path from initial state i to final state j is given a weight of $\Pr[i|\xi]\Pr[j|\xi]$. For any S such that $\sum_{i\in S}\Pr[i|\xi] \leq \frac{1}{2}$, the sum total of the weights of the paths crossing from S to $\overline{S}$ is

$$\sum_{i\in S, j\in\overline{S}} \Pr[i|\xi]\Pr[j|\xi] = \sum_{i\in S}\Pr[i|\xi]\sum_{j\in\overline{S}}\Pr[j|\xi] \geq \frac{1}{2}\sum_{i\in S}\Pr[i|\xi]$$

since $\sum_{j\in\overline{S}}\Pr[j|\xi] = 1 - \sum_{i\in S}\Pr[i|\xi]$. For any transition $t = (k,k')$, let $P(t)$ be the set of ordered pairs (l,m) such that the path from l to m contains t. Lemma 4 shows

$$\Pr[k|\xi]P_{kk'} \geq \frac{p_0}{\mathcal{D}_\xi^2}\sum_{(l,m)\in P(t)}\Pr[l|\xi]\Pr[m|\xi]$$

where p_0 denotes the smallest transition probability $\min_{i,j} P_{ij}$. It now follows that

$$\begin{aligned}\sum_{i\in S, j\in\overline{S}}\Pr[i|\xi]P_{ij} &\geq \frac{p_0}{\mathcal{D}_\xi^2}\sum_{t\in C(S)}\sum_{(l,m)\in P(t)}\Pr[l|\xi]\Pr[m|\xi] \\ &\geq \frac{p_0}{\mathcal{D}_\xi^2}\sum_{i\in S, j\in\overline{S}}\Pr[i|\xi]\Pr[j|\xi] \\ &\geq \frac{p_0}{2\mathcal{D}_\xi^2}\sum_{i\in S}\Pr[i|\xi].\end{aligned}$$

Finally

$$\Phi(H) \geq \frac{p_0}{2\mathcal{D}_\xi^2}.$$

□

We now prove the lemma used in the above analysis.

Lemma 4: Let $t = (k,k')$ be a transition from state k to state k' in $\mathcal{MC}$. Let $P(t)$ be the set of ordered pairs (l,m) such that the unique path from l to m contains t. Let p_0 denote the smallest transition probability $\min_{i,j} P_{ij}$. Then

$$\sum_{(l,m)\in P(t)}\Pr[l|\xi]\Pr[m|\xi] \leq \frac{\mathcal{D}_\xi^2}{p_0}\Pr[k|\xi]P_{kk'}.$$

□

Proof: Each $(l,m) \in P(t)$ defines a unique state $f(l,m) = k \oplus (l \oplus m)$. We show for all $(l,m) \in P(t)$

$$\Pr[l|\xi]\Pr[m|\xi] \leq \mathcal{D}_\xi^2\Pr[k|\xi]\Pr[f(l,m)|\xi]. \tag{12}$$

It follows that

$$\begin{aligned}\sum_{(l,m)\in P(t)}\Pr[l|\xi]\Pr[m|\xi] &\leq \mathcal{D}_\xi^2\Pr[k|\xi]\sum_{(l,m)\in P(t)}\Pr[f(l,m)|\xi] \\ &\leq \mathcal{D}_\xi^2\Pr[k|\xi].\end{aligned}$$

The second inequality holds because the map $f : [n]\times[n] \to [n]$ is injective and, therefore

$$\sum_{(l,m)\in P(t)}\Pr[f(l,m)|\xi] \leq \sum_{i\in[n]}\Pr[i|\xi] = 1.$$

The lemma now follows since, by definition, $P_{kk'} \geq p_0$.

We rewrite (12) as

$$\Pr[l,\xi]\Pr[m,\xi] \leq \mathcal{D}_\xi^2\Pr[k,\xi]\Pr[f(l,m)\xi]. \tag{13}$$

For any instantiation $x = (x_1,\ldots,x_n)$ of the nodes of a belief network, the joint probability of x can be factored into a product of conditional probabilities

$$\Pr[x,\xi] = \prod_{i=1}^{n}\Pr[x_i|\boldsymbol{u}_{X_i},\xi]$$

where $\boldsymbol{u}_{X_i}$ is the configuration of X_i's unobserved parents in x. Denote the binary vectors (l,ξ), (m,ξ), (k,ξ), and $(f(l,m),\xi)$ by $(l_1,\ldots,l_n,\xi)$, $(m_1,\ldots,m_n,\xi)$, $(k_1,\ldots,k_n,\xi)$, and $(f(l,m)_1,\ldots,f(l,m)_n,\xi)$. From the factorization of the joint probability distribution, we obtain

$$\begin{aligned}\Pr[l,\xi]\Pr[m,\xi] &= \Pr[l_1,\ldots,l_n,\xi]\Pr[m_1,\ldots,m_n,\xi] \\ &= \prod_{i=1}^{n}\Pr[l_i|\boldsymbol{u}_{X_i=l_i},\xi]\Pr[m_i|\boldsymbol{u}_{X_i=m_i},\xi]\end{aligned}$$

with a similar expression for $\Pr[k,\xi]\Pr[f(l,m),\xi]$. By definition of k and $f(l,m)$, either l_i and m_i have the same value, in which case k_i and $f(l,m)_i$ also have the same value l_i and m_i, or l_i and m_i differ in value, in which case k_i and $f(l,m)_i$ also differ. Thus, recalling the definition of λ_i given in Section II-A, we prove for all i

$$\begin{aligned}&\Pr[l_i|\boldsymbol{u}_{X_i=l_i},\xi]\Pr[m_i|\boldsymbol{u}_{X_i=m_i},\xi] \leq \lambda_i^2 \\ &\Pr[k_i|\boldsymbol{u}_{X_i=k_i},\xi]\Pr[f(l,m)_i|\boldsymbol{u}_{X_i=f(l,m)_i},\xi].\end{aligned}$$

Equation (13) follows from the definition $\mathcal{D}_\xi$. □

E. Analysis of the Run Time

Equation (3) gives the number of trials N required by stochastic simulation algorithms to output an estimate μ that satisfies (2). The bound on N is valid only if we sample the stationary distribution of the Markov process, and thus, we generate instantiations X with probability distribution $\Pr[X|\xi]$. We sample the stationary distribution only in the limit of an infinite simulation of $\mathcal{MC}$. For finite simulation time T, the RPD between the sampling distribution and the stationary distribution ζ is given by (11). We use methods that appear in [2], [13] to verify that if ζ satisfies

$$\zeta \leq \frac{1}{3}\epsilon\phi \tag{14}$$

(3) lies within a multiplicative constant of the number of trials required to satisfy (2). In Section III-B, we show that $\mathcal{D}$-BNRAS reduces the estimation of $\phi_p = \Pr[x_1, \ldots, x_p, \xi]$ to the estimation of n-p inferences, each with probability exceeding one half. Thus, by (14), the trials used to compute the n-p inferences are generated with $\zeta = \frac{1}{6}\epsilon$.

We obtain the number of simulations T of $\mathcal{MC}$ such that the RPD between the distribution of the trials and the distribution $\Pr[\,|\xi]$ is sufficiently small to satisfy the assumption (made in Section III-B) concerning the distribution of output instantiations.

We use the conductance bound given by Theorem 3 with (11) and let $\zeta = \frac{1}{6}\epsilon$ to obtain

$$T = 16\frac{\mathcal{D}_\xi^4}{p_0^2}\log\frac{\epsilon}{6\Pi}. \tag{15}$$

The complete run time is $(n-p)N \cdot T$, where N, which is given by (8), is the number of trials required to compute estimates of the n-p inferences produced by $\mathcal{D}$-BNRAS. Thus

$$16^2(n-p)^3\frac{\mathcal{D}_\xi^4}{\epsilon^2 p_0^2}\log\frac{\epsilon}{6\Pi}\log\frac{2}{\delta}. \tag{16}$$

IV. Complexity Results

We prove an upper bound on the conductance of the Markov process constructed previously. The upper bound yields a lower bound on the Markov process simulation time to sample the distribution $\Pr[\,|\xi]$. This lower bound represents the minimum achievable simulation time of $\mathcal{MC}$. The bound is achieved if we had available a stronger result on the convergence of Markov chains than is provided by Theorem 2.

We prove that the complexity of probabilistic inference for f-dependence networks is NP-hard for any function f such that for all n and for any constant $\alpha > 0$, $f(n) \geq 1 + \alpha$.

A. An Upper Bound on the Conductance

Previously, we proved a lower bound on the conductance $\Phi_l(H)$. We prove an upper bound

$$\Phi_u(H) = \frac{2p_0}{\mathcal{D}_\xi^{\frac{1}{8}}}$$

on the conductance. We construct a class of belief networks with this conductance.

For a fixed constant c and for any $\frac{c}{n} \leq \epsilon < 1$, we construct a class of belief networks denoted by $\mathcal{B}^\epsilon$ such that any belief network $B \in \mathcal{B}^\epsilon$ with Markov chain $\mathcal{MC}(B)$ and underlying graph H has conductance

$$\Phi(H) \leq \frac{2p_0}{\mathcal{D}^{\frac{\epsilon}{16}}}.$$

By Theorem 5, we cannot prove a lower bound on the conductance of Markov chains $\mathcal{MC}$ of belief networks that is better than $\frac{2p_0}{\mathcal{D}^{\frac{\epsilon}{16}}}$. Thus, for belief networks that are not in the class of polynomial dependence networks, the RAS does not have polynomial run time. Intuitively, the algorithm fails because the Markov chain used to construct the ζ-trial generator converges to the stationary distribution too slowly to yield a polynomial time trial generator.

We begin with a lemma.

Lemma 6: For a fixed constant c and for any $\frac{c}{n} \leq \epsilon < 1$, we can construct a class of Markov chains denoted by $\mathcal{MC}^\epsilon$ with underlying graph H and conductance

$$\Phi(H) \leq \frac{2p_0}{\mathcal{D}^{\frac{\epsilon}{16}}}$$

where

$$\mathcal{D} = \left(\frac{1+\epsilon}{1-\epsilon}\right)^{2n}.$$

Proof: $\mathcal{MC}^\epsilon$ is the Markov chain with underlying graph H constructed from an n-cube with transition probabilities $P_{vv'}$ defined below.

Let Q_k represent the vertices of H whose binary representation consists of exactly k 1's. Each vertex in Q_k has n-k edges connecting it to Q_{k+1} and k edges connecting it to Q_{k-1}. Assume that n is odd. Define S_0 to be the set

$$S_0 = \bigcup_{i=0}^{\frac{n-1}{2}} Q_i$$

and

$$\bar{S_0} = \bigcup_{i=\frac{n+1}{2}}^{n} Q_i.$$

For any $k < \frac{n-1}{2}$, arrange the transition probabilities from Q_k to Q_{k-1} and Q_{k+1} such that there is a small bias toward Q_{k-1}. Similarly, bias the transition probabilities for $k > \frac{n+1}{2}$ to favor Q_{k+1}. Specifically, for $v \in Q_k$, $k < \frac{n-1}{2}$ and for $\frac{c}{n} < \epsilon \leq 1$, let

$$P_{vv'} = \begin{cases} \frac{1-\epsilon}{4}, & \text{if } v' \in Q_{k+1}; \\ \frac{1+\epsilon}{4}, & \text{if } v' \in Q_{k-1}; \\ \frac{1}{2}, & \text{if } v = v'; \\ 0 & \text{otherwise.} \end{cases}$$

For $v \in Q_k$, $k > \frac{n+1}{2}$

$$P_{vv'} = \begin{cases} \frac{1+\epsilon}{4}, & \text{if } v' \in Q_{k+1}; \\ \frac{1-\epsilon}{4}, & \text{if } v' \in Q_{k-1}; \\ \frac{1}{2}, & \text{if } v = v'; \\ 0 & \text{otherwise.} \end{cases}$$

Let P_k be the total probability of a state in Q_k in the stationary distribution. Clearly, $P_k = \sum_{i \in Q_k} \pi_i$. Recalling that there are $k+1$ edges from Q_{k+1} to Q_k and $n-k+1$ edges from Q_{k-1} to Q_k, we derive the recurrence equation

$$P_k = 2q_{k+1}P_{k+1} + 2r_{k-1}P_{k-1} \tag{17}$$

where

$$r_{k-1} = \frac{(1-\epsilon)(n-k+1)}{4n}$$

and

$$q_{k+1} = \frac{(1+\epsilon)(k+1)}{4n}.$$

In addition, the symmetry of the construction guarantees that

$$\sum_{i=1}^{\frac{n-1}{2}} P_i = \sum_{i=\frac{n+1}{2}}^{n} P_i = \frac{1}{2}.$$

Now, consider the conductance

$$\Phi(H) = \min_S \left\{ \frac{\sum_{i \in S, j \in \bar{S}} \pi_i P_{ij}}{\sum_{i \in S} \pi_i} \right\} \leq \frac{\sum_{i \in S_0, j \in \bar{S}_0} \pi_i P_{ij}}{\sum_{i \in S_0} \pi_i}.$$

Because

$$\begin{aligned} \sum_{i \in S_0, j \in \bar{S}_0} \pi_i P_{ij} &= \sum_{i \in Q_{\frac{n-1}{2}}, j \in Q_{\frac{n+1}{2}}} \pi_i P_{ij} \\ &= \frac{1-\epsilon}{4} \sum_{i \in Q_{\frac{n-1}{2}}} \pi_i \\ &= \frac{1-\epsilon}{4} P_{\frac{n-1}{2}} \end{aligned}$$

it follows that

$$\Phi(H) \leq p_0 P_{\frac{n-1}{2}}$$

since $\frac{1-\epsilon}{4} = p_0$, the minimum transition probability in H.

Equation (17) lacks an exact solution. An upper bound on $P_{\frac{n-1}{2}}$ suffices. The recurrence equation indicates that a random walk started in state Q_k drifts to state k-1, provided $q_k < r_k$. Let m denote the value between 0 and $\frac{n-1}{2}$ for which $q_k = r_k$. We verify

$$m = \frac{1-\epsilon}{2} n \leq \frac{n-1}{2} - \frac{c}{2}$$

where the second inequality follows because, by definition, $\epsilon \geq \frac{c}{n}$. In conclusion, for $0 \leq k \leq \frac{n-1}{2}$, a random walk that begins in Q_k will drift to Q_m. From the symmetry of the cube, it is clear that the probabilities P_k increase from $k=0$ to $k=m$ and then decrease from $k=m$ to $k=\frac{n-1}{2}$.

Although it is not possible to solve (17) in closed form, the standard recurrence equation

$$P'_k = qP'_{k+1} + rP'_{k=1} \tag{18}$$

with r and q constant, has the solution

$$P'_k = A\left(\frac{r}{q}\right)^k$$

for some normalization A.

Let k_0 denote the midpoint between m and $\frac{n}{2}$—that is, $k_0 = \frac{n}{2}(1-\frac{\epsilon}{2})$. Set $q = (\frac{1+\epsilon}{4})(\frac{k_0+1}{n})$ and $r = (\frac{1-\epsilon}{4})(\frac{n-k_0+1}{n})$. Solving

$$P'_{\frac{n-1}{2}} = A\left(\frac{r}{q}\right)^{k_0}\left(\frac{r}{q}\right)^{\frac{n-1}{2}-k_0} = P'_{k_0}\left(\frac{r}{q}\right)^{\frac{n-1}{2}-k_0}.$$

Because $P'_{k_0} < 1$, we get

$$\begin{aligned} P'_{\frac{n-1}{2}} &< \left(\frac{r}{q}\right)^{\frac{n-1}{2}-k_0} \\ &< 2\left[\frac{1-\epsilon}{1+\epsilon}\right]^{\frac{n\epsilon}{8}}. \end{aligned}$$

However, by the definition of $\mathcal{D}_\xi$

$$\left[\frac{1-\epsilon}{1+\epsilon}\right]^{\frac{n\epsilon}{8}} = \frac{1}{\mathcal{D}^{\frac{\epsilon}{16}}}.$$

For $k > k_0$, the ratio of coefficients in (17) $\frac{r_k}{q_k}$ is smaller than the ratio $\frac{r}{q}$ in (18). Thus, $P'_k > P_k$ for $k_0 \leq k \leq \frac{n-1}{2}$. Finally,

$$P_{\frac{n-1}{2}} \leq \frac{2}{\mathcal{D}^{\frac{\epsilon}{16}}}$$

and

$$\Phi(H) < \frac{2p_0}{\mathcal{D}^{\frac{\epsilon}{16}}}.$$

□

Proof: Define $l = \frac{1-\epsilon}{2}$. For any $\frac{c}{4} \leq \epsilon < 1$, we construct a class of belief networks denoted by $\mathcal{B}^\epsilon$ with the following properties. For any belief network $B \in \mathcal{B}^\epsilon$, the conditional probabilities are contained in the range $[l, 1-l]$, and the Markov chain $\mathcal{MC}(B)$ is contained in $\mathcal{MC}^\epsilon$. The proof follows from Lemma 9.

To construct the appropriate belief network, we proceed as follows. The chain rule

$$\begin{aligned} \Pr[X_1, \ldots, X_n] = \Pr[X_1|X_2, \ldots, X_n] \cdot \Pr[X_2|X_3, \ldots, X_n] \\ \cdots \Pr[X_{n-1}|X_n] \cdot \Pr[X_n] \end{aligned}$$

defines a belief network on the binary-valued nodes $\{X_1, \ldots, X_n\}$ such that each node X_i has parents

$$\boldsymbol{u}_{X_i} = \{X_{i+1}, \ldots, X_n\}$$

and conditional probabilities

$$\Pr[X_i|\boldsymbol{u}_{X_i}] = \Pr[X_i|X_{i+1}, \ldots, X_n].$$

In order for the Markov chain defined by the belief network to have a the stationary distribution of $\mathcal{MC}^\epsilon$, we define for any $1 \leq k \leq n$

$$\begin{aligned} &\Pr[x_k = 1|X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_n] \\ &= \begin{cases} \frac{l}{2} & \text{if } (x_1 \cdots x_n) \in S_0; \\ \frac{1-l}{2} & \text{if } (x_1 \cdots x_n) \in \bar{S}_0 \end{cases} \end{aligned}$$

and

$$\begin{aligned} &\Pr[x_k = 0|X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_n] \\ &= \begin{cases} \frac{1-l}{2} & \text{if } (x_1 \cdots x_n) \in S_0; \\ \frac{l}{2} & \text{if } (x_1 \cdots x_n) \in \bar{S}_0. \end{cases} \end{aligned}$$

Now, we need to verify that the conditional probabilities

$$\Pr[X_k|X_{k+1},\ldots,X_n]$$

with the above definitions are contained in the interval $[l, 1-l]$. Let

$$\sum_{x_1\cdots x_{k-1}:(x_1\cdots x_n)\in S_0} \Pr[x_1,\ldots,x_{k-1}|x_{k+1},\ldots,x_n] = p$$

and

$$\sum_{x_1\cdots x_{k-1}:(x_1\cdots x_n)\in \bar{S_0}} \Pr[x_1,\ldots,x_{k-1}|x_{k+1},\ldots,x_n] = 1-p$$

where $0 \le p \le 1$. Then

$$\begin{aligned}
&\Pr[x_k|x_{k+1},\ldots,x_n]\\
&= \sum_{(x_1\cdots x_{k-1})} \Pr[x_k|x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n]\\
&\quad\cdot \Pr[x_1,\ldots,x_{k-1}|x_{k+1},\ldots,x_n]\\
&= \frac{l}{2}\cdot \sum_{x_1\cdots x_{k-1}:(x_1\cdots x_n)\in S_0} \Pr[x_1,\ldots,x_{k-1}|x_{k+1},\ldots,x_n]\\
&\quad + \frac{1-l}{2}\cdot \sum_{x_1\cdots x_{k-1}:(x_1\cdots x_n)\in \bar{S_0}}\\
&\Pr[x_1,\ldots,x_{k-1}|x_{k+1},\ldots,x_n]\\
&= \frac{l}{2}p + \frac{1-l}{2}(1-p).
\end{aligned}$$

Therefore, the conditional probabilities are contained in the interval $[l, 1-l]$. □

B. Complexity of Probabilistic Inference

Cooper [6] proves that the complexity of probabilistic inference for belief networks is NP-hard. However, it is well known that probabilistic inference for belief networks with restricted topologies, such as singly connected networks, are amenable to polynomial time algorithms [17]—that is, the complexity of the inference lies in $\boldsymbol{P}$. The following question arises: Does there exists a class of f-dependence networks such that the complexity of *exact* inference is in $\boldsymbol{P}$. Theorem 7 proves that no such class exists.

Theorem 7: For any $\alpha > 0$, the complexity of probabilistic inference for the class of belief networks characterized by dependence value $\mathcal{D}_\xi$ that satisfy $1 \le \mathcal{D}_\xi \le 1+\alpha$ is NP-hard.

Thus, we do not expect to find efficient exact algorithms for probabilistic inference under any set of restrictions placed on the range of the conditional probabilities—barring the trivial case when $\alpha = 0$ and $\mathcal{D}_\xi = 1$ occuring in the absence of conditional dependencies.

Proof: We reduce the problem of counting satisfying assignments in 3-SAT to the problem of computing probabilistic inference. The former problem is known to be ♯P-complete [21]. We use a construction first described in [6] and used there to prove that the complexity of probabilistic inference for general belief networks is ♯P-hard.

Let F be an instance of 3-SAT with variables $V = \{v_1,\ldots,v_n\}$ and clauses $C = \{c_1,\ldots,c_m\}$. The formula F defines the belief network that has binary-valued nodes $V \cup C$ and arcs directed from v_i to c_j if and only if variable v_i appears in clause c_j. Each node v_i is given a prior probability of one half of being instantiated to 0 or 1. For any clause c_j, let $j1, j2, j3$ index the three variables in V contained in c_j. The conditional probabilities associated with node c_j, which have parent nodes $\{v_{j1}, v_{j2}, v_{j3}\}$ in the belief network, are defined by

$$\Pr[c_j = 1|v_{j1}, v_{j2}, v_{j3}] = \begin{cases} \frac{1}{1+\epsilon}, & \text{if } v_{j1}, v_{j2}, v_{j3} \text{ satisfies } c_j;\\ \frac{\epsilon}{1+\epsilon}, & \text{otherwise;}\end{cases}$$

for some $0 < \epsilon < 1$. By this definition, the conditional probability $\Pr[c_j = 1|v_{j1}, v_{j2}, v_{j3}]$ has value $\frac{1}{1+\epsilon}$ if the instantiation of the variables v_{j1}, v_{j2}, v_{j3} in the formula F satisfies the clause c_j and has value $\frac{\epsilon}{1+\epsilon}$ otherwise. In addition, note that the dependence value of the belief network is given by $\mathcal{D} = \frac{1}{\epsilon^{2n}}$.

Let the vector $\mathbf{1}$ denote the instantiation $c_1 = 1,\ldots,c_m = 1$ in the belief network, and let $\mathbf{1}_i$ index the ith instantiation $c_i = 1$. For any $0 \le i < 2^n$, let the length n binary vector $\boldsymbol{v}^i$ denote the instantiation of the nodes $v_1,\ldots,v_n$ to the binary representation of i, and let $\boldsymbol{v}^i_j$ index the instantiation of v_j.

We complete the proof by showing that if we compute the inference $\Pr[\mathbf{1}]$, we may then count the number of satisfying assignments to F. By construction

$$\begin{aligned}
\Pr[\mathbf{1}] &= \sum_{i=0}^{2^n-1} \Pr[\mathbf{1}|\boldsymbol{v}^i]\Pr[\boldsymbol{v}^i]\\
&= \frac{1}{2^n}\sum_{i=0}^{2^n-1} \Pr[\mathbf{1}|\boldsymbol{v}^i].
\end{aligned}$$

From the properties of belief networks, we show that

$$\Pr[\mathbf{1}|\boldsymbol{v}^i] = \prod_{j=1}^{n} \Pr[\mathbf{1}_j|\boldsymbol{v}^i_{j1}, \boldsymbol{v}^i_{j2}, \boldsymbol{v}^i_{j3}]$$

where, as before, $j1, j2, j3$ index the three variables in V contained in c_j. Together with the definition of the conditional probabilities, we obtain

$$\Pr[\mathbf{1}|\boldsymbol{v}^j] = \frac{\epsilon^s}{(1+\epsilon)^n}$$

where s denotes the number of clauses in F that are *not* satisfied by the instantiation $\boldsymbol{v}^j$. For $0 \le s \le n$, let N_s denote the number of instantiations for which F has exactly s clauses that are not satisfied. Putting the preceding results together, we now express $\Pr[\mathbf{1}]$ by

$$\Pr[\mathbf{1}] = \frac{1}{2^n}\frac{1}{(1+\epsilon)^n}\sum_{s=0}^{n} \epsilon^s N_s. \tag{19}$$

It follows that by computing $\Pr[\mathbf{1}]$ for $n+1$ different values of ϵ such that the $n+1$ equations given by (19) are independent, and inverting the $n+1$ equations, we solve for $N_0,\ldots,N_n$. However, N_0 is the number of satisfying assignments to F. Hence, we complete the proof once we show that $n+1$ values for ϵ can be chosen such that $\mathcal{D}_0 \le \mathcal{D} \le \mathcal{D}_0 + \alpha$, given the independence of the equations. Restricting the values of ϵ to lie within the interval $[(\mathcal{D}_0+\alpha)^{-\frac{1}{2n}}, \mathcal{D}_0^{-\frac{1}{2n}}]$ satisfies the

former condition. Independence of the equations is guaranteed if the determinant of the matrix of coefficients is different from 0. However, we verify that the determinant is a multivariate polynomial in the $n+1$ different ϵ of degree $\frac{n(n+1)}{2}$. Thus, it has a finite number of roots. Since any interval in the field of rational numbers contains an infinite number of rational numbers, we always find $n+1$ values for ϵ in the interval $[(\mathcal{D}_0+\alpha)^{-\frac{1}{2n}}, \mathcal{D}_0^{-\frac{1}{2n}}]$ for which the determinant does not vanish. □

V. Conclusion

The run time of $\mathcal{D}$-BNRAS increases as the fourth power of the dependence value $\mathcal{D}_\xi$, and for belief networks with large dependence value, the run time is intractable. To address this problem, Dagum and Horvitz condition on nodes with large λ [7]. Thus, they reformulate an inference approximation in a belief network with a large dependence value into a set of inference approximations with reduced dependence value. They express the original inference as a weighted sum of subproblem inferences. They approximate these weights with logic sampling.

$\mathcal{D}$-BNRAS is ideally suited for applications where the size of the evidence set is very large—that is, applications where stochastic simulation algorithms sensitive to the size of the evidence set, such as logic sampling and likelihood weighting, perform poorly. Increases in the observed evidence ξ decreases $\mathcal{D}_\xi$ and, consequently, improves the performance of $\mathcal{D}$-BNRAS. The performance, however, is tractable only if the dependence value is polynomial. Thus, from Section II-A, $\mathcal{D}$-BNRAS is tractable on the class of belief networks with at least $n - O(\log n)$ nodes X_i having $\lambda_i = 1 + O(\frac{\log n}{n})$. For this class, when n approaches infinite, the λ_i approach 1, and the conditional probabilities $\Pr[x_i|\boldsymbol{u}_{X_i}, \xi]$ and $\Pr[\overline{x}_i|\boldsymbol{u}_{X_i}, \xi]$ approach p and 1-p for some $0 \leq p \leq 1$.

Acknowledgment

We benefited from many valuable discussions with R. Shachter. L. Dupré edited the manuscript.

References

[1] C. Berzuini, R. Bellazzi, and S. Quaglini, "Temporal reasoning with probabilities," in *Proc. 1989 Workshop Uncertainty Artificial Intell.* (Windsor, Canada), 1989, pp. 14–21.

[2] A. Broder, "How hard is it to marry at random? (On the approximation of the permanent)," in *Proc. Eighteenth ACM Symp. Theory Comput.*, 1986, pp. 50–58.

[3] R. Chavez, "Architectures and approximation algorithms for probabilistic expert systems," Ph.D. thesis, Medical Comput. Sci. Group, Stanford Univ., Stanford, CA, 1990.

[4] R. Chavez and G. Cooper, "A randomized approximation algorithm for probabilistic inference on Bayesian belief networks," *Networks*, vol. 20, pp. 661–685, 1990.

[5] ——, "A randomized approximation analysis of logic sampling," in *Proc. Sixth Conf. Uncertainty Artificial Intell.* (Cambridge, MA), 1990, pp. 130–135.

[6] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intell.*, vol. 42, pp. 393–405, 1990.

[7] P. Dagum and E. Horvitz, "Reformulating inference problems through selective conditioning," in *Proc. Eighth Conf. Uncertainty Artificial Intell.* (Stanford, CA), 1992, pp. 49–54.

[8] ——, "An analysis of Monte-Carlo algorithms for probabilistic inference," Tech. Rep. KSL-91-67, Knowledge Syst. Lab., Stanford Univ., Stanford, CA, 1991.

[9] P. Dagum and M. Luby, "Approximating probabilistic inference in Bayesian belief networks is NP-hard," Tech. Rep. KSL-91-51, Knowledge Syst. Lab., Stanford Univ., Stanford, CA, 1991.

[10] R. Fung and K. -C. Chang, "Weighing and integrating evidence for stochastic simulation in Bayesian networks," in *Uncertainty in Artificial Intelligence* **5**. Amsterdam: Elsevier, 1990, pp. 209–219.

[11] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Uncertainty in Artificial Intelligence* **2**. Amsterdam: North-Holland, 1988, pp. 149–163.

[12] M. Jerrum and A. Sinclair, "Approximating the permanent," *SIAM J. Comput.*, vol. 18, no. 6, pp. 1149–1178, 1989.

[13] M. Jerrum, L. Valiant, and V. Vazirani, "Random generation of combinatorial structures from a uniform distribution," *Theoretical Comput. Sci.*, vol. 43, pp. 169–188, 1986.

[14] R. Karp, M. Luby, and N. Madras, "Monte-carlo approximation algorithms for enumeration problems," *J. Algorithms*, vol. 10, pp. 429–448, 1989.

[15] J. Pearl, "Addendum: Evidential reasoning using stochastic simulation of causal models," *Artificial Intell.*, vol. 33, pp. 131, 1987.

[16] ——, "Evidential reasoning using stochastic simulation of causal models," *Artificial Intell.*, vol. 32, pp. 245–257, 1987.

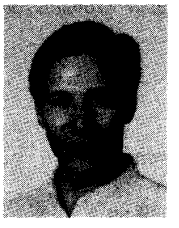
[17] ——, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[18] R. Shachter and M. Peot, "Evidential reasoning using likelihood weighting," to be published in *Artificial Intell.*

[19] ——, "Simulation approaches to general probabilistic inference on belief networks," in *Uncertainty in Artificial Intelligence* **5**. Amsterdam: Elsevier, 1990, pp. 221–231.

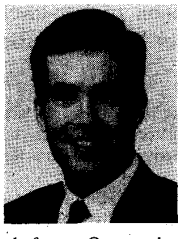
[20] A. Sinclair and M. Jerrum, "Approximate counting, uniform generation, and rapidly mixing Markov chains," *Inform. Comput.*, vol. 82, pp. 93–133, 1989.

[21] L. Valiant, "The complexity of computing the permanent," *Theoretical Comput. Sci.*, vol. 8, pp. 189–201, 1979.



Paul Dagum received the B.Sc. degree with Honors and the Gold Medal in chemical physics in 1984 from Queen's University, Canada. Funded by the Natural Sciences and Engineering Council of Canada, he received the M.Sc. degree in nuclear physics in 1986 and the Ph.D. degree in computer science in 1989, both from the University of Toronto. His doctoral dissertation was in graph theory and computational complexity.

In 1989, he spent one year as a visiting scholar at the International Computer Science Institute at the Unversity of California, Berkeley. Since 1990, he has been a postdoctoral fellow in the Section on Medical Informatics at Stanford University. His current research interests include Monte Carlo inference algorithms for Bayesian belief networks and belief network models for time-series analysis. He also studies medicine at Stanford University Medical School.



R. Martin Chavez received the A.B. degree (magna cum laude) in biochemical sciences from Harvard College in 1985, the S.M. degree in computer science from Harvard University in 1985, and the Ph.D. degree in medical informatics from Stanford University in 1991. He lives in San Francisco's SoMa district and is the founder and chief technical officer of Quorum Software Systems, which is a venture-financed start-up company that provides software solutions for MacIntosh developers who wish to make their products available on multiple platforms. Quorum has won its landmark legal battle with Apple Computer and is successfully licensing its Quorum Compatibility Engine Technology on SPARC and Silicon Graphics platforms.