The Parameterized Complexity of Approximate Inference in Bayesian Networks

Johan Kwisthout J.KWISTHOUT@DONDERS.RU.NL
Radboud University
Donders Institute for Brain, Cognition and Behaviour
Nijmegen, The Netherlands

Abstract

Computing posterior and marginal probabilities constitutes the backbone of almost all inferences in Bayesian networks. These computations are known to be intractable in general, both to compute exactly and to approximate by sampling algorithms. While it is well known under what constraints *exact* computation can be rendered tractable (viz., bounding tree-width of the moralized network and bounding the cardinality of the variables) it is less known under what constraints *approximate* Bayesian inference can be tractable. Here, we use the formal framework of *fixed-error randomized tractability* (a randomized analogue of fixed-parameter tractability) to address this problem, both by re-interpreting known results from the literature and providing some additional new results, including results on fixed parameter tractable de-randomization of approximate inference.

Keywords: Approximate inference; sampling; Bayesian networks; parameterized complexity; stochastic algorithms.

1. Introduction

Computing posterior and marginal probabilities constitutes the backbone of almost all inferences in Bayesian networks. These computations are known to be intractable in general (Cooper, 1990); moreover, it is known that *approximating* these computations is also NP-hard (Dagum and Luby, 1993). To render exact computation tractable, bounding the tree-width of the moralized network is both necessary (Kwisthout et al., 2010) and (with bounded cardinality) sufficient (Lauritzen and Spiegelhalter, 1988). For *approximate* inference, the picture is less clear, in part because there are multiple approximation strategies that all have different properties and characteristics. First of all, it matters whether we approximate marginal, respectively conditional probabilities. The approximation error can be measured absolute (independent of the probability that is to be approximated) or relative to this probability. Finally, the approximation algorithm can be deterministic (always guaranteeing a bound on the error) or randomized (return a bounded error with high probability). In this broad array there are a few (somewhat isolated) tractability results (Henrion, 1986; Dagum and Chavez, 1993; Dagum and Luby, 1993; Pradhan and Dagum, 1996; Dagum and Luby, 1997), but an overview of what can and cannot render approximate inference tractable is still lacking.

In this paper we use *fixed-error randomized tractability analysis* (Kwisthout, 2015), a recent randomized analogue of parameterized complexity analysis (Downey and Fellows, 1999), to systematically address this issue. We consider both absolute and relative approximation, using both randomized and deterministic algorithms, for the approximation of both marginal and conditional probabilities. We re-interpret old results and provide new results in terms of fixed-parameter or fixed-error tractability and intractability. In addition to identifying a number of corollaries from known results, a particular new contribution in this paper is a de-randomization of randomized marginal absolute approximation for fixed degree networks.

The remainder of this paper is structured as follows. After introducing the necessary preliminaries on Bayesian networks, approximation strategies, and parameterized computational complexity in Section 2, we give an overview of results from the literature in Section 3.1 and some new results in Section 3.2. The paper is concluded in Section 4.

2. Preliminaries

In this section we introduce notation and provide for some preliminaries in Bayesian networks, approximation algorithms, the complexity classes BPP and PP, and fixed-parameter and fixed-error tractability.

2.1 Bayesian networks

A (discrete) Bayesian network $\mathcal{B}$ is a graphical structure that models a set of discrete random variables, the conditional independences among these variables, and a joint probability distribution over these variables (Pearl, 1988). $\mathcal{B}$ includes a directed acyclic graph $\mathbf{G}_{\mathcal{B}} = (\mathbf{V}, \mathbf{A})$, modeling the variables and conditional independences in the network, and a set of parameter probabilities $\Pr$ in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network thus describes a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i))$ over its variables, where $\pi(V_i)$ denotes the parents of V_i in $\mathbf{G}_{\mathcal{B}}$. The size n of the network is the number of bits needed to represent both $\mathbf{G}_{\mathcal{B}}$ and $\Pr$. Our notational convention is to use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes. In the context of this paper we are particularly interested in the computation of marginal and conditional probabilities from the network, defined as computational problems as follows:

MPROB
Input: A Bayesian network $\mathcal{B}$ with designated subset of variables $\mathbf{H}$ and a corresponding joint value assignment $\mathbf{h}$ to $\mathbf{H}$.
Output: $\Pr(\mathbf{h})$.

CPROB
Input: A Bayesian network $\mathcal{B}$ with designated non-overlapping subsets of variables $\mathbf{H}$ and $\mathbf{E}$ and corresponding joint value assignments $\mathbf{h}$ to $\mathbf{H}$ and $\mathbf{e}$ to $\mathbf{E}$.
Output: $\Pr(\mathbf{h} \mid \mathbf{e})$.

2.2 Approximation

For a particular instance $(\mathcal{B}, \mathbf{H}, \mathbf{h})$ of MPROB, respectively $(\mathcal{B}, \mathbf{H}, \mathbf{h}, \mathbf{E}, \mathbf{e})$ of CPROB, an *approximate* result is a solution to this instance that is within guaranteed bounds of the optimal solution. We use ϵ to denote such (two-sided) bounds, and define additive approximation and relative approximation of MPROB and CPROB as follows:

AA-MPROB
Input: As in MPROB, in addition, error bound $\epsilon < 1/2$.
Output: $q(\mathbf{h})$ such that $\Pr(\mathbf{h}) - \epsilon < q(\mathbf{h}) < \Pr(\mathbf{h}) + \epsilon$.

RA-MPROB
Input: As in MPROB, in addition, error bound ϵ.
Output: $q(\mathbf{h})$ such that $\frac{\Pr(\mathbf{h})}{1+\epsilon} < q(\mathbf{h}) < \Pr(\mathbf{h}) \times (1 + \epsilon)$.

We define AA-CPROB and RA-CPROB correspondingly for the conditional case.

AA-CPROB
Input: As in CPROB, in addition, error bound $\epsilon < 1/2$.
Output: $q(\mathbf{h} \mid \mathbf{e})$ such that $\Pr(\mathbf{h} \mid \mathbf{e}) - \epsilon < q(\mathbf{h} \mid \mathbf{e}) < \Pr(\mathbf{h} \mid \mathbf{e}) + \epsilon$.

RA-CPROB
Input: As in CPROB, in addition, error bound ϵ.
Output: $q(\mathbf{h} \mid \mathbf{e})$ such that $\frac{\Pr(\mathbf{h} \mid \mathbf{e})}{1+\epsilon} < q(\mathbf{h} \mid \mathbf{e}) < \Pr(\mathbf{h} \mid \mathbf{e}) \times (1 + \epsilon)$.

In addition to the approximations above, that are guaranteed to *always* give a solution that is within the error bounds ϵ, we also define *stochastic* or randomized approximations, where this constraint is relaxed. In a stochastic approximation we have an additional *confidence* bound $0 < \delta < 1$ and we demand that the approximate result is within the error bounds ϵ with probability at least δ. If δ is polynomially bounded away from $1/2$, then the probability of answering correctly can be boosted arbitrarily close to 1 while still requiring only polynomial time.

2.3 Complexity theory

We assume that the reader is familiar with basic notions from complexity theory, such as intractability proofs, the computational complexity classes P and NP, and polynomial-time reductions. In this section we shortly review some additional concepts that we use throughout the paper, namely the complexity classes PP and BPP and some basic principles from parameterized complexity theory.

The complexity classes PP and BPP are defined as classes of decision problems that are decidable by a probabilistic Turing machine (i.e., a Turing machine that makes stochastic state transitions) in polynomial time with a particular (two-sided) probability of error. The difference between these two classes is in the bound on the error probability. *Yes*-instances for problems in PP are accepted with probability $1/2 + \epsilon$, where ϵ may depend exponentially on the input size (i.e., $\epsilon = 1/c^n$ for a constant $c > 1$). *Yes*-instances for problems in BPP are accepted with a probability that is polynomially bounded away from $1/2$ (i.e., $\epsilon = 1/n^c$). PP-complete problems, such as the problem of determining whether the *majority* of truth assignments to a Boolean formula ϕ satisfies ϕ, are considered to be intractable; indeed, it can be shown that NP $\subseteq$ PP. In contrast, problems in BPP are considered to be tractable. Informally, a decision problem Π is in BPP if there exists an efficient randomized (Monte Carlo) algorithm that decides Π with high probability of correctness. Given that the error is polynomially bounded away from $1/2$, the probability of answering correctly can be boosted to be arbitrarily close to 1 while still requiring only polynomial time.

While obviously BPP $\subseteq$ PP, the reverse is unlikely; in particular, it is conjectured that BPP = P (Clementi et al., 1998). In principle, randomized algorithms can always be de-randomized by simulating all possible combinations of stochastic transitions and taking a majority vote. If a randomized program takes n random bits in its computation, this will yield an $\mathcal{O}(2^n)$ deterministic algorithm; however, if a polynomial-time randomized algorithm can be shown to use at most $\mathcal{O}(\log|x|)$ *distinct* random bits (i.e., logarithmic in the input size $|x|$) than the de-randomization will yield a polynomial time deterministic (although not very efficient) algorithm. One can show that a $\log n$ bit random string can be amplified to a k-wise independent random n bit string in polynomial time for any fixed k (Luby, 1988), so showing that the random bits in the randomized algorithm can be constrained to be k-wise independent (rather than fully independent) is sufficient for efficient de-randomization.

Sometimes problems are intractable (i.e., NP-hard) in general, but become tractable if some *parameters* of the problem can be assumed to be small. Informally, a problem is called fixed-parameter tractable (Downey and Fellows, 1999) for a parameter κ (or a set of parameters $\{\kappa_1, \ldots, \kappa_m\}$) if it can be solved in time, exponential (or even worse) *only* in κ and polynomial in the input size $|x|$, that is, in $\mathcal{O}(f(\kappa) \cdot |x|^c)$ for a constant c and an arbitrary computable function f. In practice, this means that problem instances can be solved efficiently, even when the problem is NP-hard in general, if κ is known to be small. The complexity class FPT characterizes parameterized problems $\{\kappa\}$-Π that are fixed-parameter tractable with respect to κ. On the other hand, if Π remains NP-hard for all but finitely many values of the parameter κ, then $\{\kappa\}$-Π is para-NP-hard: bounding κ does not render the problem tractable. Conceptually in between these extremes one can place a problem $\{\kappa\}$-Π that can be solved in time $\mathcal{O}(|x|^\kappa)$; in this case, $\{\kappa\}$-Π is in the class XP.

A similar notion of parameterized tractability has been proposed for the randomized complexity class BPP (Kwisthout, 2015). A problem may need an exponential number of samples to be solved by a randomized algorithm (e.g., be PP-complete), but this number of samples may be exponential only in some parameters of the instance, while being polynomial in the input size. A parameterization of the probability of acceptance of a probabilistic Turing machine introduces the notion *fixed-error randomized tractable* for a problem with parameter κ that can be decided by a probabilistic Turing Machine in polynomial time, where the probability of error depends polynomially on the input and exponential on κ. The class FERT characterizes problems Π that can be efficiently computed with a randomized algorithm (i.e., in polynomial time, with error arbitrarily close to 0) if κ is bounded. Formally, a parameterized problem $\{\kappa\}$-$\Pi \in \mathsf{FERT}$ if there is a probabilistic Turing machine that accepts Yes-instances of Π with probability $1/2 + \min(f(\kappa), 1/|x|^c)$ for a constant c and arbitrary function $f : \mathbb{R} \to \langle 0, 1/2]$; No-instances are accepted with probability at most $1/2$. If Π is PP-hard even for bounded κ, then $\{\kappa\}$-Π is para-PP-hard. A somewhat weaker result relies on the assumption that $\mathsf{BPP} \neq \mathsf{NP}$: if Π is NP-hard for bounded κ, then $\{\kappa\}$-$\Pi \notin \mathsf{FERT}$.

While the theory of fixed parameter tractability is built on parameters that are natural numbers, the theory can easily be expanded to include monotone rational parameters (Kwisthout, 2011). We will liberally mix integer and rational parameters throughout this paper. A final observation with respect to fixed-parameter and fixed-error tractability that we want to make is that for each $\kappa_1 \subset \kappa$, if κ-$\Pi \in \mathsf{FPT}$, then κ_1-$\Pi \in \mathsf{FPT}$, and for each $\kappa \subset \kappa_2$, if κ-Π is para-NP-hard, then κ_2-Π is para-NP-hard. Similar observations can be made for FERT and para-PP-hardness.

3. Results

For exact computations, it is well known that the *tree-width* of the network and the maximum *cardinality* of the variables are crucial parameters that allow for fixed-parameter tractable computations of both marginal and posterior probabilities. For approximate computations, we identify the parameters in Table 1. The *dependence value of the evidence* (D_e) is a measure of the cumulative strength of the dependencies in the network, given a particular joint value assignment to the evidence variables (Dagum and Chavez, 1993). We can define for any node X_i its dependence strength (given evidence $\mathbf{e}$) λ_i as ${}^{u_i}\!/_{l_i}$, where l_i and u_i are the greatest, resp. smallest numbers such that

$$\forall_{x_i \in \Omega(X_i)} \forall_{\mathbf{p} \in \Omega(\pi(X_i))} l_i \leq \Pr(X_i = x_i \mid \mathbf{p}, \mathbf{e}) \leq u_i$$

The dependence value of the network (given evidence $\mathbf{e}$) is then given as $D_e = \prod_i \lambda_i$, where we assume that there are no deterministic parameters in the network. Note that this value is dependent on the evidence and can normally not be computed prior to approximation as it requires the computation of posterior probabilities that were to be approximated in the first place. The *local variance bound* (B) of a network is a measure of the representational expressiveness and the complexity of inference of the network (Dagum and Luby, 1997). It is defined similarly as the dependence value, but is not conditioned on the evidence, and a maximization rather than a product of the local values is computed. Let λ_i be ${}^{u_i}\!/_{l_i}$, where l_i and u_i are the greatest, resp. smallest numbers such that

$$\forall_{x_j \in \Omega(X_i)} \forall_{\mathbf{p} \in \Omega(\pi(X_i))} l_i \leq \Pr(X_i = x_i \mid \mathbf{p}) \leq u_i$$

The local variance bound of the network is then given as $B = \max_i(\lambda_i)$. Additional parameters that we will consider are the posterior probability $P_h = \Pr(\mathbf{h} \mid \mathbf{e})$, the prior probability of the evidence $P_e = \Pr(\mathbf{e})$, the number of evidence variables $|\mathbf{E}|$, the maximum indegree d and the maximum path length l of the network. Finally, we will also parameterize the actual approximation error ϵ.

Parameter	Meaning
D_e	*dependence value* of the evidence
B	*local variance bound* of the network
P_h	posterior probability
$\|\mathbf{E}\|$	number of evidence variables
P_e	prior probability of the evidence
d	maximum indegree of the network
l	maximum path length of the network
ϵ	absolute or relative error

Table 1: Overview of parameters used in the results

3.1 Reinterpretation of known results

In this section we review the literature on approximation algorithms with respect to parameterized complexity results. In general, if there exists a (fixed-parameter or fixed-error) tractable *relative* approximation for MPROB, respectively CPROB, there there also exists a tractable *absolute* approximation; this is due to the probabilities p that are being approximated are between 0 and 1, and thus that if $\frac{p}{1+\epsilon} < q < p \times (1 + \epsilon)$, then $p - \epsilon < q < p + \epsilon$. The intractability of absolute approximations thus implies the intractability of relative approximations. The converse does not necessarily hold, in particular if p is very small; in a sense, relative approximations are 'harder' then absolute approximations.

3.1.1 MARGINAL PROBABILITIES

For any fixed $\epsilon < 1/2^{n+1}$ it is easy to show that absolute approximate inference is PP-hard, as any approximation algorithm that is able to guarantee such a bound is able to solve MAJSAT simply by rounding the answer to the nearest $1/2^n$, this results holds for bounded d as the PP-hardness proof is constrained to networks with indegree 2 (Kwisthout, 2009). For $\epsilon \geq 1/n^c$ we can efficiently approximate the marginal inference problem absolutely by simple forward sampling (Henrion, 1986); this is a corollary of Chebyshev's inequality (Dagum and Luby, 1993). As the number of samples needed to approximate AA-MPROB by a randomized algorithm depends (only) on ϵ, this yields the following results:

Corollary 1 (Kwisthout, 2009) $\{d\}$-AA-MPROB *is* PP-*hard.*

Result 2 (Henrion, 1986) $\{\epsilon\}$-AA-MPROB $\in$ FERT.

For relative approximation, however, it is impossible to approximate RA-MPROB in polynomial time with *any* bound ϵ as this would directly indicate whether $\Pr(\mathbf{h}) = 0$ or not, and hence by a corollary of Cooper's (1990) result, would decide SATISFIABILITY. We thus have that:

Result 3 (Cooper, 1990) $\{\epsilon\}$-RA-MPROB *is* para-NP-hard.

Corollary 4 $\{\epsilon\}$-RA-MPROB $\notin$ FERT.

3.1.2 CONDITIONAL PROBABILITIES

The situation becomes more interesting when we consider computing posterior probabilities conditioned on evidence. There is a polynomial relation between the marginal and conditional relative approximation problems: As $\Pr(\mathbf{h} \mid \mathbf{e}) = \frac{\Pr(\mathbf{h},\mathbf{e})}{\Pr(\mathbf{e})}$, any guaranteed relative approximations $q(\mathbf{h})$ and $q(\mathbf{e})$ will give a relative approximation $q(\mathbf{h} \mid \mathbf{e})$ with error $(1+\epsilon)^2$ (Dagum and Luby, 1993). However, conditioning on the evidence increases the complexity of approximation in the absolute case. Dagum and Luby (1993) showed that there cannot be any polynomial time absolute approximation algorithm (unless P = NP) for any $\epsilon < 1/2$ because such an algorithm would decide 3SAT. Their proof uses a singleton evidence variable ($|\mathbf{E}| = 1$) and at most three incoming arcs per variable ($d = 3$). This intractability result implies that there cannot be a tractable relative approximation algorithm either.

Result 5 (Dagum and Luby, 1993) $\{\epsilon, d, |\mathbf{E}|\}$-AA-CPROB *and* $\{\epsilon, d, |\mathbf{E}|\}$-RA-CPROB *are* para-NP-hard.

Corollary 6 $\{\epsilon, d, |\mathbf{E}|\}$-AA-CPROB $\notin$ FERT*;* $\{\epsilon, d, |\mathbf{E}|\}$-RA-CPROB $\notin$ FERT.

In *rejection sampling* (Henrion, 1986) we basically compute a conditional probability by doing forward sampling and rejecting those samples that are not consistent with the evidence. It follows as a corollary that the number of samples needed for a guaranteed absolute error bound is proportional to the prior probability of the evidence, leading to the following result:

Result 7 (Henrion, 1986) $\{P_e, \epsilon\}$-AA-CPROB $\in$ FERT.

In so-called randomized approximation schemes one can compute bounds on the number of samples needed for efficient relative approximation of CPROB using the zero-one estimator theorem (Karp et al., 1989). As the number of samples for a particular query depends on the error ϵ, the confidence δ, and the *posterior* probability P_h, the following result follows directly from this theorem:

Result 8 (Dagum and Chavez, 1993) $\{P_h, \epsilon\}$-RA-CPROB $\in$ FERT.

These results depend on the prior probability of the evidence, respectively the posterior probability conditioned on the evidence. In a series of papers, Paul Dagum and colleagues explored efficient relative approximations of CPROB that did not depend on these probabilities, using dependence and variance bounds as explained in the introduction of this section. They proved that exact inference for every $D_e > 1$ is #P-hard (by reducing from #3SAT); this implies that we cannot have an absolute approximation for arbitrary ϵ as we could then round the approximate result in order to recover the number of satisfying 3SAT instances. They did introduce a randomized approximation scheme using ϵ, δ, and D_e as parameters. This gives the following results:

Corollary 9 (Dagum and Chavez, 1993) $\{D_e\}$-AA-CPROB *is* para-PP-hard.

Result 10 (Dagum and Chavez, 1993) $\{D_e, \epsilon\}$-RA-CPROB $\in$ FERT.

In a later paper (Dagum and Luby, 1997), the dependency on the evidence in the network in the definition of the dependence value was removed by the introduction of local bounded variance B. Computing MPROB exactly is PP-hard even when parameters are arbitrarily close to $1/2$, that is, when B is bounded; this follows as a corollary from the $\mathsf{NP}^{\mathsf{PP}}$-hardness proof of the MAP problem in Park and Darwiche (2004). The range of the posterior probabilities in this proof is such that the posterior probability of a particular query of interest in a satisfying instance is bigger then twice the probability of an unsatisfying instance, hence we cannot have a relative approximation for any $\epsilon < 1$ because that would solve SAT. In addition to this result, Dagum and Luby (1997) showed that in general $\{B\}$-AA-CPROB is NP-hard for any $\epsilon < 1/2$ and any $B > 1$. However, Dagum and Luby (1997) showed that for bounded B, $|\mathbf{E}|$, *and* ϵ one can obtain efficient Monte Carlo sampling algorithms, even when B is defined only on the query and evidence variables, i.e., $B = \max_{i \in \mathbf{H} \cup \mathbf{E}}(\lambda_i)$, rather than on the network as a whole.

Result 11 (follows from Park and Darwiche, 2004) $\{B\}$-RA-CPROB *is* NP-hard *for every* $\epsilon < 1$.

Corollary 12 $\{B\}$-RA-CPROB $\notin$ FERT *for any* $\epsilon < 1$.

Result 13 (Dagum and Luby, 1997) $\{B, |\mathbf{E}|, \epsilon\}$-RA-CPROB $\in$ FERT.

Dagum and Luby (1997) also introduced a de-randomization result; when, in addition to B and ϵ, also the *depth* of the network (i.e., the length l of the longest path in the graph) is bounded, then CPROB can be approximated with a *guaranteed* relative error ϵ in polynomial time. The de-randomized algorithm is not fixed-parameter tractable, however, as the running time is not polynomial in the input size, but exponential in the parameters. This places this parameterized problem variant in the class XP, rather than FPT.

Result 14 (Dagum and Luby, 1997) $\{B, l, |\mathbf{E}|, \epsilon\}$-RA-CPROB $\in$ XP.

3.2 New results

3.2.1 INTRACTABILITY

A number of intractability results that we list in the overview in Table 3.3 are simple corollaries from well known intractability results. For example, as it is PP-hard to absolutely approximate MPROB in general, even with bounded indegree, this implies that $\{d, P_e, |\mathbf{E}|\}$-AA-CPROB is PP-hard, as we can include an isolated evidence node with arbitrary (non-zero) prior probability without influencing the intractability result.

In this section we will explicitly prove that there cannot be an efficient absolute ϵ-approximation of the MPROB problem parameterized on the posterior probability P_h unless $\mathsf{P} = \mathsf{NP}$. We assume the existance of an algorithm A that can absolute ϵ-approximate MPROB in polynomial time when the posterior probability is larger than any number $R \leq 1 - {}^1\!/\!{}_{2^{n-1}}$, and we show that this assumption implies $\mathsf{P} = \mathsf{NP}$.

Fix R and let $q = R - {}^1\!/\!{}_{2^{n^2}}$. From an arbitrary SATISFIABILITY instance ϕ (using $\phi = \neg(x_1 \vee x_2) \wedge \neg x_3$ as a running satisfiable example) we construct a Bayesian network $\mathcal{B}_\phi$ as follows. For every literal x_i in ϕ, we construct a binary root variable X_i in $\mathcal{B}_\phi$, with uniform probability. For every logical connective in ϕ that binds the subformulae x and y, we introduce a binary variable F_{xy} with x and y as parents, and with a conditional probability distribution that matches the truth table of that connective. For example, for a disjunction connecting x_1 and x_2 we introduce a binary variable with $\Pr(F_{x_1 x_2} = \mathit{true} \mid X_1, X_2) = 1$ if either X_1 or X_2 (or both) are set to true. The top-level connective ($\wedge$ in the example formula) is denoted as V_ϕ in the network. Observe that $\Pr(V_\phi = \mathit{true}) = {}^{\#}\!/\!{}_{2^n}$, where $\#$ denotes the number of satisfying truth assignments to ϕ; more in particular, $\Pr(V_\phi = \mathit{true}) > 0$ if and only if ϕ is satisfiable. In addition, we add a binary variable H to $\mathcal{B}_\phi$, with V_ϕ as only parent, and $\Pr(H = \mathit{true} \mid V_\phi = \mathit{true}) = q + {}^1\!/\!{}_{2^{n-1}}$ and $\Pr(H = \mathit{true} \mid V_\phi = \mathit{false}) = q$. Figure 1 gives the resulting network for the running example $\phi = \neg(x_1 \vee x_2) \wedge \neg x_3$.

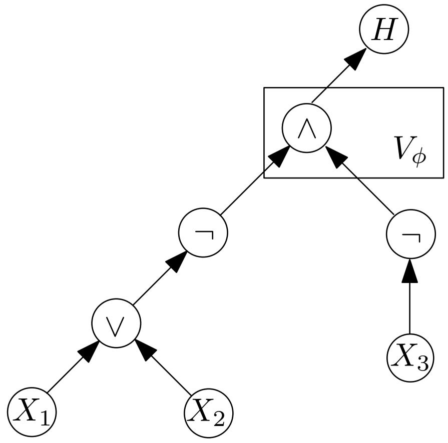

Figure 1: The Bayesian network $\mathcal{B}_\phi$ corresponding to the example SATISFIABILITY instance $\phi = \neg(x_1 \vee x_2) \wedge \neg x_3$

Theorem 15 *If there exists a polynomial-time* AA-MPROB *algorithm for any posterior probability larger than* $R \leq 1 - {}^1\!/_{2^{n-1}}$*, then* $\mathsf{P} = \mathsf{NP}$.

Proof Let ϕ be a SATISFIABILITY instance with n variables and let $\mathcal{B}_\phi$ be the Bayesian network constructed from ϕ as discussed above. Obviously this construction takes polynomial time. Let $q = R - {}^1\!/_{2^{n^2}}$. We then have that $\Pr(H = \mathit{true}) = (q \times 1) + ((q + {}^1\!/_{2^{n-1}}) \times 0) = q$ if ϕ is not satisfiable, and $\Pr(H = \mathit{true}) \geq (q \times 1 - {}^1\!/_{2^n}) + ((q + {}^1\!/_{2^{n-1}}) \times {}^1\!/_{2^n}) = q + {}^2\!/_{2^{n^2}}$ if ϕ is satisfiable. Let $\epsilon < {}^1\!/_{2^{n^2}}$. For any range $R \subseteq [0, 1 - {}^1\!/_{2^{n-1}}]$, the existance of a polynomial-time ϵ-approximation of $\Pr(H = \mathit{true})$ would immediately decide whether ϕ was satisfiable. We conclude that there cannot exist an polynomial-time algorithm approximating AA-MPROB for any posterior probability $P_h \in [0, 1 - {}^1\!/_{2^{n-1}}]$ unless $\mathsf{P} = \mathsf{NP}$. ■

Corollary 16 $\{P_h\}$-AA-MPROB $\notin$ FERT.

3.2.2 DE-RANDOMIZATION

We will show that $\{d, \epsilon\}$-AA-MPROB is fixed-parameter tractable by de-randomizing Result 2. Let $\{\mathcal{B}, \mathbf{H}, \mathbf{h}, \epsilon\}$ be an instance of AA-MPROB. A simple forward sampling algorithm that would compute a single sample for $\mathbf{H}$ using a number of random bits which is linear in the input size n, i.e., $\mathcal{O}(n)$[1]. This algorithm would be run N times to achieve a randomized approximation algorithm, and by Result 2 N would be exponential only in ϵ, and polynomial in the input size. We can derandomize this algorithm by simulating it for every combination of values of the random bits and making a majority vote, but this would give an exponential time algorithm as the number of random bits is $\mathcal{O}(n)$. If $\mathcal{O}(\log n)$ *distinct* random bits would suffice, however, we could effectively simulate the randomized algorithm in polynomial time, as the $\mathcal{O}(\log n)$ random bits can be amplified in polynomial time to a k-wise independent $\mathcal{O}(n)$-bit random string for every fixed k (Luby, 1988).

In general, we cannot assume k-wise independence in the random bits. As the forward sampling algorithm samples values for variables from their CPTs, given a (previously sampled) joint value assignment to the parents of these variables, the random numbers need to be fully independent for all parents of these variables; that is, if the maximum number of parents any variable has in the network is d (i.e., the maximal indegree of the network) then the random numbers need to be d-wise independent. So, we can effectively de-randomize a simple forward sampling randomized approximation algorithm for AA-MPROB to a tractable deterministic algorithm if d is bounded. This gives us a $\{d, \epsilon\}$ fixed parameter tractable algorithm for AA-MPROB.

Result 17 $\{d, \epsilon\}$-AA-MPROB $\in$ FPT.

As a corollary from Results 7 and 17, we can also de-randomize $\{P_e, \epsilon\}$-AA-CPROB to prove the following result:

Result 18 $\{P_e, d, \epsilon\}$-AA-CPROB $\in$ FPT.

[1] Note that we defined the input size as the total number of bits needed to describe both the graph and the CPTs.

3.3 Overview

The results from the previous sub-sections are summarized in Table 3.3. The table shows there are a number of combinations of parameters that can render randomized approximation of posterior probabilities (either by a sampling approach or a Monte Carlo Markov Chain approach) tractable, typically bounding both the quality of approximation ϵ, some property of the probability distribution (like the local variance), and/or some property of the evidence (such as the number of evidence variables or the prior probability of the evidence). From a complexity-theoretic perspective the contrast between the two de-randomization results is interesting. Although it is widely assumed that BPP $=$ P, in the parameterized world we need an extra constraint on d to go from a fixed-error randomized tractable algorithm for $\{\epsilon\}$-AA-MProb to a fixed-parameter tractable one; the de-randomization of $\{B, \epsilon, |\mathbf{E}|\}$-CProb with the added constraint on l, however, leads to an $\mathcal{O}(n^{f(\kappa)})$ algorithm, i.e., an XP-algorithm, rather than an FPT-algorithm.

	MProb		CProb	
Parameters	*abs. approx.*	*rel. approx.*	*abs. approx.*	*rel. approx.*
$\{\epsilon\}$	$\in$ FERT [2]	$\not\in$ FERT [4]	$\not\in$ FERT [6]	$\not\in$ FERT [6]
$\{d\}$	PP-hard [1]	PP-hard [1]	PP-hard [1]	PP-hard [1]
$\{d, \epsilon\}$	$\in$ FPT [17]	$\not\in$ FERT [4]	$\not\in$ FERT [6]	$\not\in$ FERT [6]
$\{d, \epsilon, \|\mathbf{E}\|\}$	N/A	N/A	$\not\in$ FERT [6]	$\not\in$ FERT [6]
$\{d, P_e, \|\mathbf{E}\|\}$	N/A	N/A	PP-hard [1]	PP-hard [1]
$\{P_e, \epsilon\}$	N/A	N/A	$\in$ FERT [7]	?
$\{P_e, d, \epsilon\}$	N/A	N/A	$\in$ FPT [18]	?
$\{P_h\}$	$\not\in$ FERT [16]	$\not\in$ FERT [16]	$\not\in$ FERT [16]	$\not\in$ FERT [16]
$\{P_h, \epsilon\}$	$\in$ FERT [8]	$\in$ FERT [8]	$\in$ FERT [8]	$\in$ FERT [8]
$\{B\}$	N/A	N/A	$\not\in$ FERT [12]	$\not\in$ FERT$^{*)}$ [12]
$\{B, \epsilon, \|\mathbf{E}\|\}$	N/A	N/A	$\in$ FERT [13]	$\in$ FERT [13]
$\{B, l, \epsilon, \|\mathbf{E}\|\}$	N/A	N/A	$\in$ XP [14]	$\in$ XP [14]
$\{D_e\}$	N/A	N/A	PP-hard [9]	PP-hard [9]
$\{D_e, \epsilon\}$	N/A	N/A	$\in$ FERT [10]	$\in$ FERT [10]

Table 2: Overview of fixed-parameter and fixed-error tractability and intractability results. The number between the brackets refers to the number of the result. '$\in$ FPT' denotes that the parameterized problem $\{\kappa\}$-$\Pi \in$ FPT, '$\in$ XP' denotes that the parameterized problem $\{\kappa\}$-$\Pi \in$ XP, '$\in$ FERT' denotes that $\{\kappa\}$-$\Pi \in$ FERT, '$\not\in$ FERT' denotes that $\{\kappa\}$-$\Pi \not\in$ FERT unless BPP $=$ NP, 'PP-hard' denotes that $\{\kappa\}$-Π is para-PP-hard, and '?' denotes that this problem is open. N/A means that the parameterization is not applicable for the marginal case where we have not evidence variables. *): This result holds for every $\epsilon < 1$. For $\epsilon \geq 1$ we currently have no relative approximation results.

4. Conclusion

In this paper we investigated the parameterized complexity of approximate Bayesian inference, both by re-interpretation of known results in the formal framework of fixed-error randomized complexity theory, and by adding a few new results. The overview in Table 3.3 gives an insight in what constraints can, or cannot, render approximation tractable. These constraints are notably different from the traditional constraints (treewidth and cardinality) that are necessary and sufficient to render exact computation tractable. In future work we wish to extend this line of research to other problems in Bayesian networks, notably the MAP problem.

Acknowledgments

The author thanks Cassio P. De Campos and Todd Wareham for useful discussions.

References

A. Clementi, J. Rolim, and L. Trevisan. Recent advances towards proving P=BPP. In E. Allender, editor, *Bulletin of the EATCS*, volume 64. EATCS, 1998.

G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990.

P. Dagum and R. Chavez. Approximating probabilistic inference in Bayesian belief networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):246–255, 1993.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

P. Dagum and M. Luby. An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence*, 93:1–27, 1997.

R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, Berlin, 1999.

M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Proceedings of the Second International Conference on Uncertainty in Artificial Intelligence*, pages 149–163, 1986.

R. Karp, M. Luby, and N. Madras. Monte-carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10:429–448, 1989.

J. Kwisthout. *The Computational Complexity of Probabilistic Networks*. PhD thesis, Faculty of Science, Utrecht University, The Netherlands, 2009.

J. Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52(9):1452 – 1469, 2011.

J. Kwisthout. Tree-width and the computational complexity of MAP approximations in Bayesian networks. *Journal of Artificial Intelligence Research*, 53:699–720, 2015.

J. Kwisthout, H. L. Bodlaender, and L. C. van der Gaag. The necessity of bounded treewidth for efficient inference in Bayesian networks. In H. Coelho, R. Studer, and M. Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 237–242. IOS Press, 2010.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50(2):157–224, 1988.

M. Luby. Removing randomness in parallel computation without a processor penalty. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS'88)*, pages 162–173, 1988.

J. D. Park and A. Darwiche. Complexity results and approximation settings for MAP explanations. *Journal of Artificial Intelligence Research*, 21:101–133, 2004.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Palo Alto, CA, 1988.

M. Pradhan and P. Dagum. Optimal Monte Carlo estimation of belief network inference. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 446–453. Morgan Kaufmann Publishers Inc., 1996.