The Complexity of MAP Inference in Bayesian Networks Specified Through Logical Languages

Denis D. Mauá
Universidade de São Paulo
São Paulo, Brazil
denis.maua@usp.br

Cassio P. de Campos
Queen's University Belfast
Belfast, United Kingdom
c.decampos@qub.ac.uk

Fabio G. Cozman
Universidade de São Paulo
São Paulo, Brazil
fgcozman@usp.br

Abstract

We study the computational complexity of finding maximum a posteriori configurations in Bayesian networks whose probabilities are specified by logical formulas. This approach leads to a fine grained study in which local information such as context-sensitive independence and determinism can be considered. It also allows us to characterize more precisely the jump from tractability to NP-hardness and beyond, and to consider the complexity introduced by evidence alone.

1 Introduction

What is the complexity of finding most probable explanations in Bayesian networks? Conventional wisdom says that such a search is an NP-complete problem; if the network has polytree structure, matters improve as the problem then lies in P. But suppose the network is specified so that every probability distribution is a deterministic function, except for a few selected marginal probabilities. Complexity should clearly depend on which deterministic functions can be used when specifying probabilities. So, the question is: what is the complexity of finding most probable explanations (MPE) and maximum a posteriori (MAP) configurations, as parameterized by the language that is allowed in specifying the deterministic functions? This is the question we explore in this paper. More precisely, we look at deterministic functions that are specified through logical formulas. In doing so, we are inspired by previous work that focused solely on inference problems [3].

It is surprising that even for propositional languages we find our question to produce a nuanced set of results. We spend most of this paper considering combinations of conjunction, disjunction, and negation, and we are able to pinpoint the crucial role of disjunction in complexity jumps.

Even though most existing results on complexity of Bayesian networks assume that probability distributions are given as tables of rational numbers [4; 9; 10; 15; 18], in practical situations the specification language does matter. For instance, one may have algorithmic gains by employing Noisy-OR gates and the like [7; 15], by resorting to supermodular functions [13; 6] or to relational languages that make symmetries explicit [5; 16], or even by considering regular structures [21]. By parameterizing the complexity of finding MPE and MAP by specification language, we seek to understand this interplay between expressivity and computational cost. We show that restrictions on language can sometimes bypass the effect of graph topology and can lead to tractable inferences even in networks of high treewidth. On the other hand, we show that MAP inference is intractable even for some very simple (sub Boolean) languages.

We start with some necessary background in Section 2, and then examine the complexity of MPE and MAP inferences in propositional languages (Section 3). Section 4 examines template (relational) languages. Section 5 concludes the paper.

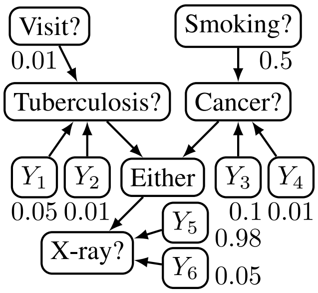
2 MPE and MAP, parameterized by language

A Bayesian network consists of a directed acyclic graph whose nodes are random variables $X_1, \dots, X_n$ and a collection of conditional probability distributions $\mathbb{P}(X_i|\mathrm{pa}(X_i))$, one distribution for each variable/node in the graph. Roughly speaking, the *treewidth* of a graph measures its tree-likeness, with a minimum treewidth of one characterizing trees. The inferential complexity of Bayesian networks has been shown to be strongly tied to the treewidth when the probability distributions are specified using tables of rationals [9; 10]. On the other hand, there is an extensive literature on exploiting local structure such as determinism, symmetry and parameter sharing to speed up inference [13; 7; 6; 16; 21].

In order to circumvent the treewidth barrier, and formalize the inferential complexity in terms of local structure, Cozman and Mauá [3] recently proposed using Bayesian networks over Boolean variables (i.e., taking values $\{0, 1\}$) whose conditional probability distributions $\mathbb{P}(X_i|\mathrm{pa}(X_i))$ are deterministic functions specified as logical equivalences $X_i \Leftrightarrow \phi_i(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$, where ϕ_i is a well-formed formula in some (fixed) language $\mathcal{L}$.[1] They showed that such networks can represent any network (by augmenting the model), and that new classes of Bayesian networks with tractable marginal inference can be found by restricting the language in which conditional probabilities are specified.

Given a Bayesian network over variables $X_1, \dots, X_n$, a set of so-called MAP variables $\mathbf{M} \subseteq \{X_1, \dots, X_n\}$ and an

[1] Following [3], we use variables and their corresponding propositions interchangeably.



$$T \Leftrightarrow (Y_1 \wedge V) \vee (Y_2 \wedge \neg V)$$

$$C \Leftrightarrow (Y_3 \wedge S) \vee (Y_4 \wedge \neg S)$$

$$E \Leftrightarrow T \vee C$$

$$X \Leftrightarrow (Y_5 \wedge E) \vee (Y_6 \wedge \neg E)$$

Figure 1: A fragment of the Asia network; numbers by root nodes convey the probability that the variable in the node is true; logical equivalences refer to abbreviated names.

event $\mathbf{E}$ formed by a conjunction of assignments $X_e = x_e$, where X_e is a variable not in $\mathbf{M}$, the maximum a posteriori configuration problem is to compute

$$\max_{\mathbf{m}} \mathbb{P}(\mathbf{M} = \mathbf{m}, \mathbf{E})\,. \tag{1}$$

The decision variant of the problem (denoted by MAP) is to decide whether (1) exceeds a given rational number. When $\mathbf{M}$ and $\mathbf{E}$ span all variables, the problem is known as the most probable explanation problem (denoted by MPE). We refer to $\mathbf{E}$ as the *evidence*. When variables are Boolean, we call an assignment $X_e = 1$ a *positive evidence*, and an assignment $X_e = 0$ a *negative evidence*.

The aim of this work is to investigate the complexity of MAP and MPE inferences parameterized by the language in which conditional probabilities are specified. We denote by $\mathsf{Prop}(O_1, O_2, \dots)$ the set of propositional well-formed formulas with propositions and Boolean operators $O_1, O_2, \dots$. For instance, $\mathsf{Prop}(\wedge, \neg)$ is the set propositional sentences with conjunction and negation (hence all propositional sentences). We denote by $(\neg)$ the negation operator that can only be applied to root variables (i.e., variables with no parents).

For instance, consider the celebrated QMR-DT network (meaning "Quick Medical Reference, Decision-Theoretic" network) [20]. This is a large network with a bipartite graph where root nodes are "diseases" and non-root nodes are "findings", all of them represented by Boolean variables. There are about 600 diseases in QMR-DT, each one of them associated with a marginal probability assessment, and about 4000 findings, each one of them associated with a Noisy-OR gate. That is, QMR-DT is clearly a network specified within the framework described, using $\mathsf{Prop}(\wedge, \vee)$ as specification language. In fact, QMR-DT is a perfect example of a network that has very large treewidth (there are cliques with more than 150 variables in efficient triangulations), and that can be handled by smart exploitation of the specification language [7].

Now consider another example, where the point is to mention fact that any Bayesian network can be specified with $\mathsf{Prop}(\wedge, \neg)$. Figure 1 shows a small fragment of the popular Asia network [11], entirely encoded with $\mathsf{Prop}(\wedge, \neg)$ plus some auxiliary variables Y_i.

Given a class $\mathcal{B}$ of Bayesian networks specified with a language $\mathcal{L}$, we denote by $\mathsf{MAP}_d(\mathcal{L})$ the set of all MAP *decision* problems whose inputs are networks in $\mathcal{B}$. We denote by $\mathsf{MAP}_d^+(\mathcal{L})$ the subclass of problems in $\mathsf{MAP}_d(\mathcal{L})$ with positive evidence (i.e., assignments $X_e = 1$); similarly, we denote by $\mathsf{MAP}_d^0(\mathcal{B})$ the subclass of MAP problems with no evidence (i.e., $\mathbf{E} = \emptyset$). Clearly, $\mathsf{MAP}_d(\mathcal{L})$ is at least as hard as $\mathsf{MAP}_d^+(\mathcal{L})$, which is at least as hard as $\mathsf{MAP}_d^0(\mathcal{L})$.

We adopt analogous definitions for MPE: $\mathsf{MPE}_d(\mathcal{L})$, $\mathsf{MPE}_d^+(\mathcal{L})$ and $\mathsf{MPE}_d^0(\mathcal{L})$ are the class of MPE problems with Bayesian network specified with $\mathcal{L}$ and arbitrary, positive and no evidence, respectively. Because MPE is a subcase of MAP, computing a MAP problem in a specific language is at least as hard as computing the MPE restricted to that language. Intractability of MPE implies intractability of MAP; tractability of MAP implies tractability of MPE.

3 Complexity Results

This section contains complexity results of MAP and MPE inference parameterized by some propositional languages. We assume familiarity with NP-completeness theory. A probabilistic Turing machine is a nondeterministic Turing machine that accepts a string iff the majority of the paths accepts. The class PP contains languages that can be decided by a probabilistic Turing machine in polynomial time. The class $\mathsf{NP}^{\mathsf{PP}}$ contains languages that can be decided in polynomial time by a nondeterministic Turing machine with an oracle PP.

We focus on MPE problems in Section 3.2, then move to MAP problems in Section 3.3. Before we walk into a maze of theorems about complexity, let us pause and overview our main results.

3.1 Overview

Shimony [19] was the first to show that MPE is NP-complete for table-based Bayesian networks even when there is no evidence [19], and therefore NP-complete for $\mathsf{Prop}(\wedge, \neg)$ (without evidence). Shimony's proof can be easily adapted to show that $\mathsf{MPE}_d^0(\mathsf{Prop}(\wedge, \vee))$ is also NP-complete by noting that conditional probabilities can be specified using only conjunctions and disjunctions.[2] We prove in Theorem 1 that NP-completeness also obtains for the very simple language $\mathsf{Prop}(\wedge)$. The idea of the proof is to use negative evidence to obtain disjunction from conjunction. If we only consider positive evidence, then MPE becomes polynomial, even if we allow negation of root nodes (Theorem 2). But if we allow disjunction with positive evidence (Theorem 3) or conjunction with negative evidence, then MPE is already NP-complete, because we can use the evidence to obtain the other operator. That is: either conjunction or disjunction is fine, but both they take us to NP-completeness. Finally, we show, by a different technique, that exclusive-OR $\oplus$ also leads to tractability (Theorem 4).

We then move to MAP. Because MAP is generally much harder than MPE, finding a language for which MAP problems are as hard as MPE problems can be seen as a positive

[2] Shimony uses "probability drain nodes" to virtually set an evidence into the network by assigning a very small probability to disagreeing assignments. Drain nodes can be specified as the disjunction of their parents and "fresh" auxiliary nodes with uniform probability.

result. Here proofs are more involved, as there are few $\mathsf{NP}^{\mathsf{PP}}$-complete problems in the literature.

Park and Darwiche were the first to examine the complexity of MAP inference in Bayesian networks, and concluded that the problem is $\mathsf{NP}^{\mathsf{PP}}$-complete for table-based specifications [14]. A corollary of their result is that $\mathsf{MAP}_d(\mathsf{Prop}(\wedge,\neg))$ is $\mathsf{NP}^{\mathsf{PP}}$-complete. Their proof can be modified to show that the more restricted variant $\mathsf{MAP}_d^+(\mathsf{Prop}(\wedge,\vee,(\neg)))$ is already $\mathsf{NP}^{\mathsf{PP}}$-complete.

In Theorem 5 we prove our most challenging result, namely, that $\mathsf{MAP}_d^+(\mathsf{Prop}(\vee))$ is *already* an $\mathsf{NP}^{\mathsf{PP}}$-complete problem. We then show that a language with only conjunction also produces a MAP problem that is $\mathsf{NP}^{\mathsf{PP}}$-complete if we allow negated evidence (Theorem 6). These results suggest again that allowing conjunction and disjunction adds considerable difficulty to inference, even we if need to use evidence to build the other operator.

We then move to $\mathsf{Prop}(\wedge)$ with positive evidence. The complexity of MAP is generally understood as the complexity of an MPE problem where each evaluation requires the complexity of marginal inference. This intuition has been corroborated by some complexity results. For example, for networks specified through $\mathsf{Prop}(\wedge,\neg)$, the decision version of marginal inference is PP-complete, MPE is NP-complete and MAP is $\mathsf{NP}^{\mathsf{PP}}$-complete. Following the same intuition, we expect the complexity of $\mathsf{MAP}_d^0(\mathsf{Prop}(\wedge))$ to be the complexity of MPE with an oracle for inference in $\mathsf{Prop}(\wedge)$. This MPE problem is in P; also, the inference problem is in P [3]; hence we expect $\mathsf{MAP}_d^0(\mathsf{Prop}(\wedge))$ to be in P. However, we have the surprising result that MAP in such networks is PP-hard (Theorem 7). Note that we only prove hardness; perhaps $\mathsf{MAP}_d^0(\mathsf{Prop}(\wedge))$ is $\mathsf{NP}^{\mathsf{PP}}$-complete, or PP-complete. We leave the question open.

3.2 MPE

Theorem 1. $\mathsf{MPE}_d(\mathsf{Prop}(\wedge))$ *is* NP*-complete.*

Proof. Pertinence is immediate (it follows for any MPE problem). To prove hardness, we use a reduction from $\mathsf{VERTEX\ COVER}$:

> **Input:** A graph $G=(V,E)$ and an integer k.
> **Question:** Is there a set $C\subseteq V$ of cardinality at most k such that each edge in E is incident to at least one node in C?

Given an instance of $\mathsf{VERTEX\ COVER}$, construct a Bayesian network as follows. For each node v in V, add a variable X_v with $\mathbb{P}(X_v=1)=3/4$. For each edge $e=(u,v)$ in E, add a node X_e whose parents are X_u and X_v, and specify $X_e \Leftrightarrow X_u \wedge X_v$. Intuitively, the nodes $X_v=0$ if node v is in a vertex cover and $X_v=1$ otherwise, whereas $X_e=0$ if the edge e is covered by some of its endpoints, otherwise $X_e=1$. The Bayesian network on the right in Figure 2 is obtained from a $\mathsf{VERTEX\ COVER}$ problem with the graph on the left. Consider an assignment $\mathbf{v}$ to the node variables $\{X_v : v\in V\}$, and denote by $\mathbf{v}(X_v)$ the assignment of variable X_v, $v\in V$. Let $C(\mathbf{v})=\{v:\mathbf{v}(X_v)=0\}$ be the node set represented by $\mathbf{v}$. By construction, if $C(\mathbf{v})$ is not a vertex cover (i.e., if

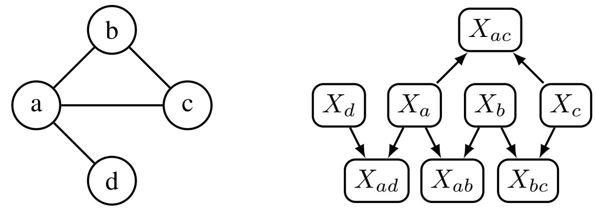


Figure 2: A Bayesian network (on the right) that solves $\mathsf{VERTEX\ COVER}$ with the graph on the left.

there is X_e such that both its parents are assigned the value one in $\mathbf{v}$), then $\mathbb{P}(\mathbf{V}=\mathbf{v},\{X_e=0,e\in E\})=0$, since $\mathbb{P}(X_e=0|X_u=\mathbf{v}(X_u),X_v=\mathbf{v}(X_v))=0$ for some $e=(u,v)\in E$. On the other hand, if $C(\mathbf{v})$ is a vertex cover then $\mathbb{P}(\mathbf{V}=\mathbf{v},\{X_e=0,e\in E\})=\prod_{v\in C(\mathbf{v})}(1-\mathbb{P}(X_v=1))\prod_{v\notin C(\mathbf{v})}\mathbb{P}(X_v=1)=3^{n-|C|}/4^n$, where $n=|V|$. Hence, there is an edge cover C of size $|C|\leq k$ if and only if $\max_{\mathbf{v}}\mathbb{P}(\mathbf{V}=\mathbf{v},\{X_e=0,e\in E\})\geq 3^{n-k}/4^n$. $\square$

Theorem 2. $\mathsf{MPE}_d^+(\mathsf{Prop}(\wedge,(\neg)))$ *and* $\mathsf{MPE}_d^0(\mathsf{Prop}(\vee))$ *are in* P*.*

Proof. For solving $\mathsf{MPE}_d^+(\mathsf{Prop}(\wedge,(\neg))$, propagate the evidence by making all ancestors of evidence nodes take on value one (true), which is the only configuration assigning positive probability. Now, for both $\mathsf{MPE}_d^+(\mathsf{Prop}(\wedge,(\neg))$ and $\mathsf{MPE}_d^0(\mathsf{Prop}(\vee))$, the procedure is as follows. Assign values of the remaining root nodes as to maximize their marginal probability independently (i.e., for every non-determined root node X select $X=1$ if and only if $\mathbb{P}(X=1)\geq 1/2$). Assign the remaining internal nodes the single value which makes their probability non-zero. This can be done in polynomial time and achieves the maximum probability. $\square$

Theorem 3. $\mathsf{MPE}_d^+(\mathsf{Prop}(\vee))$ *is* NP*-complete.*

Proof. The proof is similar to the proof of Theorem 1. Given an instance of $\mathsf{VERTEX\ COVER}$ with graph $G=(V,E)$ and integer k, construct a Bayesian network containing nodes X_v, $v\in V$, associated with the probabilistic assessment $\mathbb{P}(X_v=1)=1/4$ and nodes $X_e, e=(u,v)\in E$, associated with the logical equivalence $X_e\Leftrightarrow X_u\vee X_v$. Let $C(\mathbf{v})=\{v:\mathbf{v}(X_v)=1$. Then $\mathbb{P}(\mathbf{V}=\mathbf{v},\{X_e=1,e\in E\})=\prod_{v\in C(\mathbf{v})}\mathbb{P}(X_v=1)\prod_{v\notin C(\mathbf{v})}(1-\mathbb{P}(X_v=1))=3^{n-|C|}/4^n\geq 3^{n-k}/4^n$ if and only if $C(\mathbf{v})$ is s vertex cover of cardinality at most k. $\square$

Theorem 4. $\mathsf{MPE}_d^+(\mathsf{Prop}(\oplus))$ *is in* P*.*

Proof. The operation $\oplus$ is supermodular, hence the logarithm of the joint probability is also supermodular and the MPE problem can be solved efficiently [13]. $\square$

3.3 MAP

Theorem 5. $\mathsf{MAP}_d^+(\mathsf{Prop}(\vee))$ *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete.*

Proof. Consider the following $\mathsf{NP}^{\mathsf{PP}}$-complete problem, known as $\mathsf{EMAJ(3)SAT}$ [12; 14]:

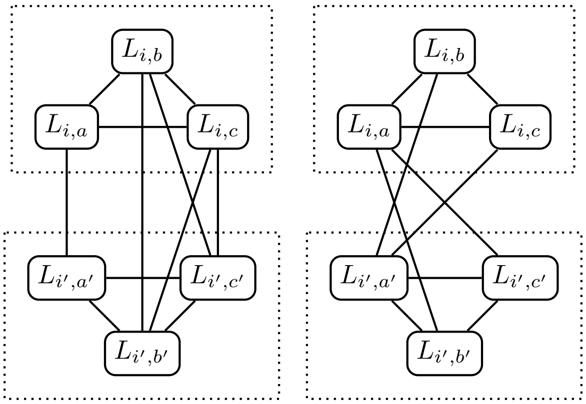


Figure 3: Two examples of gadgets used in the proof of Theorem 5. The graph on the left represents two clauses that share a Boolean variable in the literals $L_{i,a} = \neg L_{i',a'}$. On the right, a similar construct but where the literals have the same sign $L_{i,a} = L_{i',a'}$. Each connection means that the endpoint nodes cannot both assume the state 0.

> **Input:** A 3-CNF formula $\psi(X_1, \ldots, X_n)$ and an integer k.
> **Question:** Is there an assignment to $X_1, \ldots, X_k$ such that the majority (that is, at least $1/2 \cdot 2^{n-k}$) of the assignments to $X_{k+1}, \ldots, X_n$ satisfies ψ?

To prove the desired result, we first prove $\mathsf{NP}^{\mathsf{PP}}$-hardness of an intermediate problem, which is a variant of EMAJ(3)SAT. Given a formula in CNF over $X_1, \ldots, X_k$, we say that an assignment respects the *1-in-3 rule* if at most one literal in each a clause takes on true (and to satisfy the formula, at least one needs to be true). The E(1-in-3)SAT problem is:

> **Input:** Given a 3-CNF formula $\phi(X_1, \ldots, X_n)$ with m clauses and integer k.
> **Question:** Is there an assignment to $X_1, \ldots, X_k$ such that at least $1/2^{1+m}$ of the assignments to $X_{k+1}, \ldots, X_n$ respect the 1-in-3 rule and satisfies ϕ?

We will derive a parsimonious reduction from EMAJ(3)SAT to E(1-in-3)SAT. Let m' be the number of clauses of a EMAJ(3)SAT instance with formula ψ over n variables, and denote by $R(l_1, l_2, l_3)$ the relation that is true if and only if exactly one of its literals l_1, l_2, l_3 is true. Clauses of E(1-in-3)SAT will be represented by such relation as follows: each clause $l_{i,1} \vee l_{i,2} \vee l_{i,3}$ of EMAJ(3)SAT can be written as the set $R(\neg l_{i,1}, a_i, b_i) \wedge R(l_{i,2}, b_i, c_i) \wedge R(\neg l_{i,3}, c_i, d_i) \wedge R(a_i, c_i, e_i) \wedge R(a_i, c_i, e_i)$ of clauses of the E(1-in-3)SAT problem, where a_i, b_i, c_i, d_i, e_i are new Boolean variables only appearing in these relations. The number of satisfying assignments to the latter expression equals the number of satisfying assignments to $l_{i,1} \vee l_{i,2} \vee l_{i,3}$. Consider the formula ϕ obtained by replacing each of the m' clauses in ψ by such corresponding expression. The E(1-in-3)SAT problem is that of deciding whether there is an assignment to $X_1, \ldots, X_k$ such that at least $1/2^{1+m}$ of the 2^{m+n-k} assignments to $X_{k+1}, \ldots, X_n$ and to $(A_i, B_i, C_i, D_i, E_i)_{i=1,\ldots,m'}$ (where $m = 5m'$) respect the 1-in-3 rule and satisfy ϕ. Since there is a one-to-one correspondence between satisfying solutions of both problems, and the E(1-in-3)SAT instance contains $5m'$ more variables than the EMAJ(3)SAT instance such that $1/2^{1+m} = \frac{1/2 \cdot 2^{n-k}}{2^{5m'+n-k}}$, where the numerator is 1/2 of the assignments of the EMAJ(3)SAT instance and the denominator is the number of assignments of the E(1-in-3)SAT instance, we obtain that E(1-in-3)SAT is $\mathsf{NP}^{\mathsf{PP}}$-hard.

We now prove the desired result (i.e., hardness of $\mathsf{MAP}_d^+(\mathsf{Prop}(\vee))$) by a reduction from E(1-in-3)SAT. Let $R(l_{i,1}, l_{i,2}, l_{i,3})$ represent the clause i of the E(1-in-3)SAT formula ϕ ($i \in \{1, \ldots, m\}$) and let $X_j \in \{0, 1\}$ be the j-th Boolean variable ($j \in \{1, \ldots, n\}$). Let $X_{i,r}$ be the variable appearing in the literal $l_{i,r}$. For each literal $l_{i,r}$ appearing in ϕ, create a root node $L_{i,r}$ in the network (there might be multiple root nodes corresponding to the same Boolean variable; nodes will be denoted with uppercase L and literals of the formula with lowercase l). Define $\mathbb{P}(L_{i,r} = 1) = \varepsilon$ and interpret the state 0 of the node $L_{i,r}$ to mean that the literal $l_{i,r}$ is set to *true* as for the satisfiability of the formula ϕ. Now, for each pair of literals $(l_{i,r}, l_{i',r'})$ such that $X_{i,r} = X_{i',r'}$ and $l_{i,r} = \neg l_{i',r'}$, include a disjunction evidence node $E_{i,r,i',r'}$ with positive observation on it (that is, observe state 1). This will eventually force the values of nodes $L_{i,r}$ and $L_{i',r'}$ not to be 0 (that is, true) at the same time, as desired. We use the notation $E_{i,r,i',r'}$ to indicate the logical relation implicitly, that is, $E_{i,r,i',r'} \Leftrightarrow L_{i,r} \vee L_{i',r'}$. Now, for each clause i, build three disjunction evidence nodes $E_{i,1,i,2}, E_{i,1,i,3}, E_{i,2,i,3}$ and set the observation to be positive on them (state 1), so no two literal nodes in a clause can be 0 at the same time (this implies each clause will have at most one literal node with state 0, enforcing the 1-in-3 rule). Examples of this "clause gadget" can be seen inside the dashed rectangles in Figure 3.

With the transformation so far, we already know that every configuration for the root nodes that does not respect the 1-in-3 rule will lead to a joint probability of zero, so we conclude that there are at most 4^m configurations with non-zero probability. We can also state that an arbitrary assignment $x_1, \ldots, x_n$ of the Boolean variables in the E(1-in-3)SAT problem satisfying ϕ and respecting the 1-in-3 rule will have a corresponding configuration for all root nodes in the network by setting the literal nodes according to the literals in the formula (and by using the satisfying assignment $x_1, \ldots, x_n$). Such configuration of the root nodes will have probability $\alpha = (1 - \varepsilon)^m \cdot \varepsilon^{2m}$, because exactly one literal node per clause will be set to state 0 and two nodes to state 1. Recall that we interpret the state zero 0 of a literal node in the network as defining such literal as true (thus state 1 means to assign false to the literal). Now take a configuration of the root nodes in the network corresponding to a situation where at least one clause of ϕ is not satisfied, that is, where all literal nodes corresponding to a clause in ϕ are set to state 1. This configuration will have probability at most $\beta = (1 - \varepsilon)^{m-1} \cdot \varepsilon^{2m+1}$.

The issue still to be addressed in this reduction is that there are many configurations for the root nodes that are logically incompatible but may reach probability α (for example, take two clauses $R(l_{1,x}, l_{1,y}, l_{1,z})$ and $R(l_{2,t}, l_{2,w}, l_{2,z})$ where $X_{1,z} = X_{2,z}$ and $l_{1,z} = l_{2,z}$ and set the nodes

$L_{1,x} = 0$ and $L_{2,z} = 0$, while leaving $L_{1,y} = L_{1,z} = L_{2,t} = L_{2,w} = 1$; this is possible according to the transformation but does not correspond to a valid assignment of the original Boolean variables because $L_{1,z}$ should be equal to $L_{2,z}$). To avoid such scenarios (as well as scenarios with other incompatible literal nodes about the same Boolean variable), we define additional disjunction nodes in the network. For each two clauses i and i' sharing a variable $X_{i,a} = X_{i',a'}$ such that $l_{i,a} = \neg l_{i',a'}$, create disjunction evidence nodes $E_{i,b,i',b'}, E_{i,b,i',c'}, E_{i,c,i',b'}, E_{i,c,i',c'}$ which will guarantee that if $L_{i,b} = 0$ or $L_{i,c} = 0$ then $L_{i',a'} = 0$ and $L_{i,a} = 1$ in *any satisfying assignment*, and analogously we have that if $L_{i',b'} = 0$ or $L_{i',c'} = 0$ then $L_{i,a} = 0$ and $L_{i',a'} = 1$ in any satisfying assignment (we do not care about configurations related to non-satisfying assignments because they have very low probability, as we will see). This is illustrated on the left side of Figure 3. For each two clauses i and i' sharing a variable $X_j = X_{i,a} = X_{i',a'}$ appearing in the literals $l_{i,a} = l_{i',a'}$, include disjunction evidence nodes $E_{i,a,i',b'}, E_{i,a,i',c'}, E_{i,b,i',a'}, E_{i,c,i',a'}$, which will force literals $L_{i,a}$ and $L_{i',a'}$ to be compatible in both clauses in any satisfying assignment. This is illustrated on the right side of Figure 3. With such additional constraints, configurations of the root nodes that are incompatible get zero probability.

Because at most 4^m configurations can have non-zero probability (in fact, because of the compatibility constraints we have included, possibly less than 4^m root node configurations can have non-zero probability), the sum of all configurations that correspond to assignments not satisfying ϕ is at most $4^m \cdot \beta$. By choosing a value of ε such that $4^m \cdot \beta = 4^m \cdot (1-\varepsilon)^{m-1} \cdot \varepsilon^{2m+1} < 2^{-m-1}\alpha = 2^{-m-1}(1-\varepsilon)^m \cdot \varepsilon^{2m} \iff 2^{3m+1} < \frac{1}{\varepsilon} - 1 \iff \varepsilon < \frac{1}{2^{3m+1}+1}$, we can tell exactly the number of satisfying assignments of ϕ as a multiple of α with a error less than 2^{-m-1} (all non satisfying assignments summed together will not change this value enough to reach the next multiple of $2^{-m-1}\alpha$).

Now we choose as MAP variables all of those related to literals that contain variables $X_1, \ldots, X_k$, that is, $\mathbf{M} = \{L_{i,a} \mid X_{i,a} \in \{X_1, \ldots, X_k\}, i \in \{1, \ldots, m\}\}$. We pose the query to the MAP problem of whether there is a configuration $\mathbf{m}$ for the MAP variables such that $\mathbb{P}(\mathbf{m}) \geq 2^{-m-1+n-k}\alpha$, which happens if and only if given Boolean variables $X_1, \ldots, X_k$ fixed according to $\mathbf{m}$ (which represents a valid assignment otherwise $\mathbb{P}(\mathbf{m}) = 0$), at least 2^{-m-1} assignments to $X_{k+1}, \ldots, X_n$, satisfy ϕ. The answer is affirmative if and only if E(1-in-3)SAT answer is affirmative, hence $\mathsf{MAP}^+_d(\mathsf{Prop}(\vee))$ is $\mathsf{NP}^{\mathsf{PP}}$-hard (pertinence in $\mathsf{NP}^{\mathsf{PP}}$ is immediate as it is contained in the class of MAP_d problems). ☐

Theorem 6. $\mathsf{MAP}_d(\mathsf{Prop}(\wedge))$ *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete.*

Proof. $\mathsf{MAP}^+_d(\mathsf{Prop}(\vee))$, which is $\mathsf{NP}^{\mathsf{PP}}$-complete by Theorem 5, can be reduced to $\mathsf{MAP}_d(\mathsf{Prop}(\wedge))$ by inverting the meaning of all nodes' states and by applying de Morgan's law to the network. Hence, all disjunctions become conjunctions and evidence nodes become all negated, hence $\mathsf{MAP}_d(\mathsf{Prop}(\wedge))$ is $\mathsf{NP}^{\mathsf{PP}}$-hard. As MAP_d is itself a problem in $\mathsf{NP}^{\mathsf{PP}}$, completeness follows trivially. ☐

Theorem 7. $\mathsf{MAP}^0_d(\mathsf{Prop}(\wedge))$ *and* $\mathsf{MAP}^0_d(\mathsf{Prop}(\vee))$ *are* PP*-hard.*

Proof. We reduce MAJ-2MONSAT to $\mathsf{MAP}^0_d(\mathsf{Prop}(\wedge))$, which is PP-complete [17]:

> **Input:** A 2-CNF formula $\phi(X_1, \ldots, X_n)$ with m clauses where all literals are positive.
> **Question:** Does the majority of the assignments to $X_1, \ldots, X_n$ satisfy ϕ?

The transformation is as follows. For each Boolean variable X_i, build a root node such that $\mathbb{P}(X_i = 1) = 1/2$. For each clause C_j with literals x_i and x_k (note that literals are always positive), build a conjunction node E_j with parents X_i and X_k, that is, $E_j \Leftrightarrow X_i \wedge X_k$. Now set all non-root nodes to be MAP nodes, that is, $\mathbf{M} = \{E_1, \ldots, E_m\}$.

Suppose that variables in $\mathbf{M}$ are chosen to be $\mathbf{m}$ where at least one of them is set to state 1. This implies that both parents of this conjunction node must be set to state 1 too, and thus the joint probability $\mathbb{P}(\mathbf{m}) \leq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ (as there are two root nodes with their states fixed to 1). So any MAP configuration $\mathbf{m}$ where not all nodes are set to state 0 will have probability inferior to $\frac{1}{4} < \frac{1}{2}$. On the other hand, take the MAP configuration $\mathbf{m} = \mathbf{0}$ (vector equality). In this case, by construction, $\mathbb{P}(\mathbf{m})$ is exactly the number of satisfying assignments of ϕ: just interpret each configuration of the root nodes such that $X_i = 0$ means that the Boolean variable X_i is set to *true*, while node $X_i = 1$ means *false*. Because the MAP nodes are set to 0, at least one of its parents need to be set to state 0, which means that the configuration satisfies the formula ϕ. As the MAP value is computed as a sum over all possible satisfying assignments, the obtained probability is exactly the percentage of satisfying assignments of ϕ. Hence it is enough to check whether exists $\mathbf{m}$ with $\mathbb{P}(\mathbf{m}) > \frac{1}{2}$. There is such a MAP configuration (which can only happen when all MAP nodes are set to state 0) if and only if the majority of the assignments satisfies ϕ. In order to prove the result for $\mathsf{MAP}^0_d(\mathsf{Prop}(\vee))$, we just use the very same reasoning with $\vee$ in E nodes (MAP variables will be set to 1). ☐

4 Going relational

In our quest for increased expressivity, we can imagine moving from random variables to parameterized random variables; that is, instead of just considering e.g. proposition Rich representing whether a particular individual is rich, we might consider e.g. relation Rich($\mathfrak{x}$) with $\mathfrak{x}$ ranging over a domain of individuals, and expressing the wealth condition of each individual. The idea is that we now have parameterized random variables [16], and our models are templates that get instantiated by specific sets of individuals. This allows us to investigate the complexity of networks where many nodes share the same conditional probabilities. There are indeed many such relational extensions of Bayesian networks [8; 5]. Suppose that we allow each node of our networks to be associated with a relation; given a domain, we can produce all possible groundings, and build a grounded Bayesian network by connecting groundings if their corresponding relations are connected. In this paper a domain is always a finite set, so we always produce a finite Bayesian network through grounding.

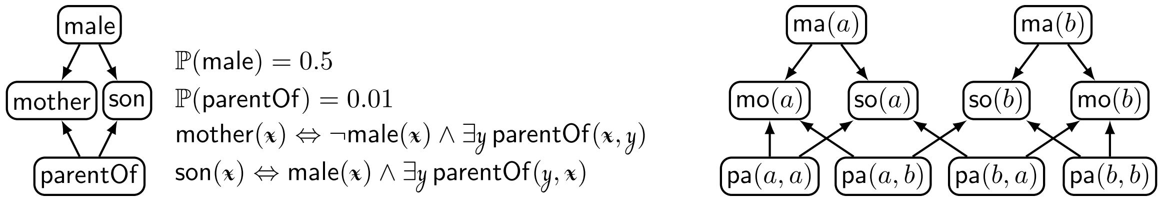


Figure 4: Example of relational Bayesian network and its grounding with $\mathcal{D} = \{a, b\}$.

The models discussed in the previous paragraph are essentially identical to *relational Bayesian networks* [8], in that each node is associated with a relation, and the semantics is given by grounding into Bayesian networks. The reader can find more details about the specific features of relational Bayesian networks, and their semantics, in the literature [8]. Our only change with respect to Jaeger's original relational Bayesian networks is that we adopt again Cozman and Mauá's scheme for probability specification [3]. That is, we let each root node be associated with a probability $\mathbb{P}(\mathsf{r}) = \alpha$, where α is a rational number in $[0, 1]$, and each non-root node be associated with a logical equivalence $\mathsf{r}(x_1, \dots, x_k) \Leftrightarrow \phi(x_1, \dots, x_k)$, where $\phi(x_1, \dots, x_k)$ is a well-formed formula in some fragment of first-order logic, with the restriction that all and only logical variables associated with r are free variables in ϕ. See Figure 4 for an example.

Suppose we have as input a directed acyclic graph and associated probabilities and equivalences, plus a domain (a finite list of individuals), and evidence (a conjunction of assessments for groundings). By MPE we now mean the same MPE problem as before, but applied to the grounded network, with the given evidence. And similarly for the MAP problem.

However, when we focus on relational languages we can consider additional questions. For instance, in practice it may be that the size of the relational model is small, whereas the size of the evidence, or the size of the domain, is large. So we may ask: what is the complexity of MPE when the relational model is fixed, but the evidence or domain grow? We denote by DMPE the MPE problem where the size of the relational model is bounded (and assumed constant), and only evidence and domain size may change. This sort of analysis concentrates on the so-called data complexity [3].

We start with a very simple relational language [3], obtained by restricting the popular description logic DL-Lite [2]. Denote by DLDiet the language consisting of the set of formulas recursively defined so that any unary relation is a formula, $\neg\mathsf{r}(x)$ is a formula for any unary relation r, $\phi \wedge \varphi$ is a formula when both ϕ and φ are formulas, and $\exists y\, \mathsf{r}(x, y)$ is a formula for a binary relation r. Note that because this language contains $\mathsf{Prop}(\wedge)$, it follows that $\mathsf{MPE}_d(\mathsf{DLDiet})$ is NP-complete. However, by fixing the size of the relation model, we obtain:

Theorem 8. $\mathsf{DMPE}_d(\mathsf{DLDiet})$ *is in P.*

Proof. The proof is rather similar to the proof of polynomial-time inference by Cozman and Mauá [3]. By construction the grounded Bayesian network consists of disconnected slices, whose variables can be selected independently. Each slice can be solved by brute force conditional on the values of the existential relations; this takes time exponential in the size of the relational network that is considered bounded (hence constant time). For each assignment of values to existential relations, the values of the binary relations can be decided greedily by inspecting the marginal probability (e.g., if the existential is set to true and the probability of the corresponding relation is less than 1/2, assign an arbitrary parent relation to true and the remaining to false). The whole process takes polynomial time. □

Consider now the language EL [3], inspired by the homonymous popular description logic [1] as defined as follows. Any unary relation is a formula in EL, $\phi \wedge \varphi$ is a formula when both ϕ and φ are formulas, and $\exists y\, \mathsf{r}(x, y) \wedge \mathsf{s}(y)$ is a formula when r is a binary relation and s is a unary relation. For this language, a fixed bound on the size of the relational model does not reduce complexity.

Theorem 9. $\mathsf{DMPE}_\mathsf{d}(\mathsf{EL})$ *is NP-complete.*

Proof. Pertinence can be shown by solving the MPE in the grounded network, an MPE problem. To show hardness, consider a VERTEX COVER problem in a graph with nodes $V = \{v_1, \dots, v_n\}$ and edges $E = \{e_1, \dots, e_m\}$ given by its incidence matrix C: C is an n-by-m matrix such that $A_{ij} = 1$ if node i is incident upon edge j and zero otherwise. Note that $\sum_i A_{ij} = 2$ for any j. Obtain a square matrix A' by augmenting A with zeros (i.e., if $n > m$ add $n - m$ zero-filled columns; if $n < m$ add $m - n$ zero-filled rows). Construct the relational Bayesian network specified by $\mathbb{P}(\mathsf{r}(x, y)) = 1/2$, $\mathbb{P}(\mathsf{s}(y)) = 1/4$, $\mathsf{t}(x) \Leftrightarrow \exists y\, \mathsf{r}(x, y) \wedge \mathsf{s}(y)$. Let $N = \max\{n, m\}$, $\mathcal{D} = \{1, \dots, N\}$. A grounding $\mathsf{r}(j, i)$ represents the incidence of node v_i upon an edge e_j. A grounding $\mathsf{s}(i)$ indicates whether node v_i is selected in an vertex cover, while a grounding $\mathsf{t}(j)$ indicates whether edge e_j is covered by the vertex cover. Set $\mathbf{E} = \{\mathsf{r}(i, j) = 1 : A'_{ij} = 1\} \cup \{\mathsf{r}(i, j) = 0 : A'_{ij} = 0\} \cup \{\mathsf{t}(j) : \sum_i A'_{ij}/2 = 1\}$. Hence, the evidence ensures that $\mathsf{t}(j) = 1$ if and only if $\mathsf{s}(i_1) = 1$ or $\mathsf{s}(i_2) = 1$, where v_{i_1} and v_{i_2} are the nodes incident upon e_j with $j \leq m$; for $j > m$ any assignment to $\mathsf{s}(i)$ satisfies the evidence $\mathsf{t}(j) = 0$. The joint probability of the Bayesian network is $\mathbb{P}(\mathsf{s}(1), \dots, \mathsf{s}(N), \mathbf{E}) = \prod_{(v_{i_1}, v_{i_2}) \in E} \mathbf{I}(\mathsf{s}(i_1) \vee \mathsf{s}(i_2)) \prod_{i=1}^{N} \mathbb{P}(\mathsf{s}(i))$, where $\mathbf{I}$ is the indicator function which returns one if its argument evaluates to true and zero otherwise. Consider an assignment $s_1, \dots, s_N$ to $\mathsf{s}(1), \dots, \mathsf{s}(N)$ and let $C = \{v_i : i \leq n, s_i = 1\}$. If C is a vertex cover then $\mathbb{P}(\mathsf{s}(1) = s_1, \dots, \mathsf{s}(N) = s_N, \mathbf{E}) = 3^{N-|C|}/4^N$; if C is not an vertex cover then $\mathbb{P}(\mathsf{s}(1) = s_1, \ldots, \mathsf{s}(N) = s_N, \mathbf{E}) = 0$. Hence $\max_{s_1,\ldots,s_N} \mathbb{P}(\mathsf{s}(1) = s_1, \ldots, \mathsf{s}(N) = s_N, \mathbf{E}) \geq 3^{N-k}/4^N$ if and only if there is a vertex cover of cardinality at most k. $\square$

The proof above makes use of negated evidence on binary relations; the complexity of the problem with positive evidence (that is, $\mathsf{DMPE}^+_d(\mathsf{EL})$) remains open.

5 Conclusion

In this paper we have presented complexity results for MPE and MAP problems in Bayesian networks, as parameterized by the language in which probabilities are specified. Complexity results for these problems have focused on graph topology, but it has been long known that restrictions on parameter specification lead to distinct computational behavior. Our analysis offers an initial formalization of this perception. Overall we find that disjunction is the key source of computational intractability; as long as disjunction can be represented, complexity jumps considerably.

There are many possible extensions of this work. We only barely touched relational models; there are many languages that deserve attention. Issues such as separating the effect of data size and domain size from network size are important and should also be examined.

Acknowledgments

The first author was partially supported by the São Paulo Research Foundation (FAPESP) through the grant 2013/23197-4. The third author was partially supported by CNPq.

References

[1] F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 325–330, 2003.

[2] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proceedings of the AAAI Conference*, pages 602–607, 2005.

[3] F. G. Cozman and D. D. Mauá. Bayesian networks specified using propositional and relational constructs: Combined, data, and domain complexity. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[4] C. P. de Campos. New complexity results for MAP in Bayesian networks. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2100–2106, 2011.

[5] L. Getoor, N. Friedman, D. Koller, A. Pfeffer, and B. Taskar. Probabilistic relational models. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[6] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):271–279, 1989.

[7] D. Heckerman. A tractable inference algorithm for diagnosing multiple diseases. In *In Proceedings of the 5th Conference on Uncertainty in Artificial Intelligence*, 1989.

[8] M. Jaeger. Relational Bayesian networks. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 266–273, 1997.

[9] J. Kwisthout. Treewidth and the computational complexity of MAP approximations. In *Proceedings of the 7th European Workshop on Probabilistic Graphical Models*, pages 271–285, 2014.

[10] J. Kwisthout, Hans L. Bodlaender, and L. C. van der Gaag. The necessity of bounded treewidth for efficient inference in Bayesian networks. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 237–242, 2010.

[11] S. Lauritzen and D. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 50(2):157–224, 1988.

[12] M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

[13] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

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

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[16] D. Poole. First-order probabilistic inference. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.

[17] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

[18] S. E. Shimony and C. Domshlak. Complexity of probabilistic reasoning in directed-path singly-connected bayes networks. *Artificial Intelligence*, 151:213–225, 2003.

[19] S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, 1994.

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

[21] A. Weller and T. Jebara. On MAP inference by MWSS on perfect graphs. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.