Structure Approximation of Most Probable Explanations in Bayesian Networks

Johan Kwisthout

Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands, j.kwisthout@donders.ru.nl

Abstract. Typically, when one discusses approximation algorithms for (NP-hard) problems (like Traveling Salesperson, Vertex Cover, Knapsack), one refers to algorithms that return a solution whose *value* is (at least ideally) close to optimal; e.g., a tour with almost minimal length, a vertex cover of size just above minimal, or a collection of objects that has close to maximal value. In contrast, one might also be interested in approximations algorithms that return solutions that *resemble* the optimal solutions, i.e., whose *structure* is akin to the optimal solution, like a tour that is almost similar to the optimal tour, a vertex cover that differs in only a few vertices from the optimal cover, or a collection that is similar to the optimal collection. In this paper, we discuss structure-approximation of the problem of finding the most probable explanation of observations in Bayesian networks, i.e., finding a joint value assignment that *looks like* the most probable one, rather than has an *almost as high value*. We show that it is NP-hard to obtain the value of *just a single variable* of the most probable explanation. However, when partial orders on the values of the variables are available, we can improve on these results.

1 Introduction

A key computational problem in Bayesian networks [17] is the computation of the *most probable explanation* (MPE) of a set of observed phenomena; i.e., given a Bayesian network whose variables are partitioned into an *evidence* set $\mathbf{E}$ with observed joint value assignment $\mathbf{e}$ and an *explanation* set $\mathbf{M}$, determine the joint value assignment $\mathbf{m}$ to the explanation set $\mathbf{M}$ such that $\Pr(\mathbf{M} = \mathbf{m}, \mathbf{E} = \mathbf{e})$ is maximal. This problem, also called Bayesian abduction, is a key component in many decision support systems like [15, 21], in many Bayesian models of cognition, for example intention recognition [2] or recipient design [22], as well as in various models of sociological [19] or economical [8] processes.

Unfortunately, computing the MPE is in general NP-hard [12, 3, 18] and remains NP-hard when the most probable explanation is to be *approximated* rather than exactly computed. In particular, it is NP-hard to find a joint value assignment whose probability is within a fixed ratio of the most probable joint value assignment [1] and it is even NP-hard to find a joint value assignment that has a non-zero probability [12]. However, these formal notions of approximation focus

on the *value* of the explanation, i.e., the goal is to find an explanation whose *probability* is 'close' to the probability of the most probable explanation. Sometimes we may not be primarily interested in finding explanations with an almost-as-high probability, but rather in explanations that are *quite similar* to the most probable explanation, that is, they *look like* the most probable explanation. For example, in cognitive science, one's goal is to describe, model, and predict human cognition. In such applications it is conceivable that we are most interested in approximating structure, rather than value [16]; we will refer to this notion of approximation as *structure approximation* (note that the term 'structure' does *not* refer to the graphical structure (i.e., the arcs) of the network, but to the structure of the joint value assignments).

Preferably, of course, in many domains we would like to have an approximation that both resembles the optimal solution *and* have an almost-as-high probability [4]. While it may well be the case that 'good' value approximations sometimes have a similar structure as the optimal solution, this need not be the case, as we will show in Subsection 2.3.

Structure approximation has its roots in computational complexity theory [11, 6]. The relevance of structure approximation, in particular in the context of the so-called Coherence Problem, was first suggested by Millgram [16] and extensively studied in Hamilton et al. [9] and Van Rooij et al. [23]. In this paper we further build on this work and discuss structure approximations of MPE. In the remainder of this paper, we will discuss some relevant preliminaries and definitions in Bayesian networks and structure approximation in Section 2. In Section 3 we focus on structure-approximating MPE. We discuss the computational complexity of structure approximation of MPE in general in Subsection 3.1, and the effect of having an *ordering* of the variables in Subsection 3.2. In Section 4 we conclude this paper.

2 Preliminaries

In this section we introduce Bayesian networks and, more in particular, the problem of finding the most probable explanation (MPE) for a subset of variables in the network, given observations for the other variables. For more background, the reader is referred to textbooks as [17, 10] and overview papers as [14, 12]. Furthermore, we introduce a formal definition of structure approximation, as presented in [9]. We assume that the reader is familiar with basic notions in complexity theory, such as the classes P and NP and NP-hardness proofs; for more background, we refer to [7].

2.1 Bayesian networks and the MPE problem

A Bayesian or probabilistic network $\mathcal{B}$ is a graphical structure that models a set of stochastic variables, the conditional independencies among these variables, and a joint probability distribution over these variables. $\mathcal{B}$ includes a directed

acyclic graph $\mathbf{G}_{\mathcal{B}} = (\mathbf{V}, \mathbf{A})$, modeling the variables and conditional independencies 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 models 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 $\mathbf{G}_{\mathcal{B}}$. We will 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. We will use $\mathbf{E}$ to denote a set of evidence nodes, i.e., a set of nodes for which a particular joint value assignment $\mathbf{e}$ is observed; likewise, we will use $\mathbf{M}$ to denote a set of nodes for which the explanation is sought. We will sometimes write $\Pr(\mathbf{x})$ as a shorthand for $\Pr(\mathbf{X} = \mathbf{x})$ if no ambiguity can occur. We denote with $\Omega(X)$ the set of all values that X can take; $\Omega(\mathbf{X})$ is defined analogously for sets of variables.

Among other computational problems defined on Bayesian networks, one particularly interesting problem for many applications is the problem of determining the *most probable explanation* for some observations, i.e., the most probable joint value assignment to a subset of variables in the network, given evidence for the other variables[1]. This problem is formally defined as follows [12].

MPE
Instance: A probabilistic network $\mathcal{B} = (\mathbf{G}_{\mathcal{B}}, \Pr)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ with a joint value assignment $\mathbf{e}$, and an explanation set $\mathbf{M}$.
Output: $\operatorname{argmax}_{\mathbf{m}} \Pr(\mathbf{m}, \mathbf{e})$, i.e., the most probable joint value assignment $\mathbf{m}$ to the nodes in $\mathbf{M}$ and evidence $\mathbf{e}$, or the designated symbol $\perp$ if $\Pr(\mathbf{m}, \mathbf{e}) = 0$ for every joint value assignment $\mathbf{m}$ to $\mathbf{M}$.

MPE is intractable in general; to be precise, the problem is $\mathsf{FP}^{\mathsf{NP}}$-complete and has an NP-complete decision variant [12, 18].

2.2 Structure approximation

The notion of a structure approximation is typically captured using a *solution distance function*, a metric associated with each optimization problem relating candidate solutions with the optimal solution [9]. Let Π be a optimization problem with instance x, let $\mathit{cansol}(x)$ denote a function returning candidate solutions to x, with $\mathit{optsol}(x)$ denoting a function returning the *optimal* solution[2] to x. For any $y, y' \in \mathit{cansol}(x)$, let $d(y, y')$ be the distance between y and y' as defined by d. As d is a metric, the following properties hold for all $a, b, c \in \mathit{cansol}(x)$:

[1] If we have only partial evidence, i.e., the network is partitioned into variables for which the explanation is sought, evidence variables, and other variables that constitute neither evidence nor explanation, then the problem generalized to a Partial (or Marginal) MAP problem. The intractability results presented here generalize also to Partial MAP.

[2] Or, in case of a draw, one of the optimal solutions.

1. $d(a, a) = 0$
2. if $a \neq b$, $d(a, b) > 0$
3. $d(a, b) = d(b, a)$
4. $d(a, b) + d(b, c) \geq d(a, c)$

Typically, for many problems Π, d might correspond to the *Hamming distance* or *edit distance* between two candidate solutions: the number of elements in which the candidate solutions differ, or the number of operations needed to transform one candidate solution into another. We define a h/d-structure approximation of Π as follows:

Definition 1 ([9]). *Given an optimization problem Π, a solution-distance function d, and a non-decreasing function $h : \mathbb{N} \to \mathbb{N}$, an algorithm A is a polynomial-time h/d-structure approximation algorithm if for every instance x of Π, $d(A(x), optsol(x)) \leq h(|x|)$, and A runs in time polynomial in $|x|$.*

Similarly, we define an *expected* h/d-structure approximation of Π as follows:

Definition 2. *Given an optimization problem Π, a solution-distance function d, and a non-decreasing function $h : \mathbb{N} \to \mathbb{N}$, an algorithm A is a polynomial-time expected h/d-structure approximation algorithm if, for a random instance x of Π, the expected distance $E(d(A(x), optsol(x))) \leq h(|x|)$, and A runs in time polynomial in $|x|$.*

2.3 Value versus structure approximation

Possibly counter to intuition, a "good" value approximation is not necessarily a "good" structure approximation and vice versa. As an example, consider the Bayesian network in Figure 1 with binary variables $V, X_1, \ldots, X_n$, a uniform probability distribution for the variables X_1 to X_n, and the following conditional probability distribution for V:

$$\Pr(V = \textsc{true} \mid X_1, \ldots, X_n) = \begin{cases} 1 & \text{if } \forall_i X_i = \textsc{true} \\ 1 - \epsilon & \text{if } \forall_i X_i = \textsc{false} \\ 0 & \text{otherwise} \end{cases}$$

Note that the most probable explanation for the observation $V = \textsc{true}$ would be the explanation where all variables X_i are set to TRUE, and the second most probable explanation where all variables X_i are set to FALSE. Any non-zero value approximation thus would yield an explanation with a completely different structure than the most probable explanation. On the other hand, any explanation that has a similar structure (i.e., differ in only few variables) would have a probability of zero.

3 Structure approximation of MPE

Let $cansol(\mathcal{B}, \mathbf{e})$ denote the set of explanations (i.e., joint value assignments to $\mathbf{M}$) of a Bayesian network $\mathcal{B}$ with observed evidence $\mathbf{e}$, with $optsol(\mathcal{B}, \mathbf{e})$ as the

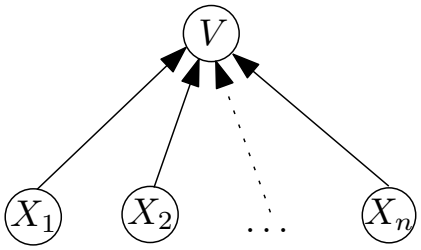

Fig. 1. Example network with distinct structure and value approximations

most probable explanation, i.e., the joint value assignment to $\mathbf{M}$ with the highest joint probability. We define the structure distance function $d_H(\mathbf{m}, \mathit{optsol}(\mathcal{B}, \mathbf{e}))$ as the Hamming distance between explanation $\mathbf{m} \in \mathit{cansol}(\mathcal{B}, \mathbf{e})$ and the most probable explanation.

In the remainder of this paper, we consider h to be a function taking an MPE instance $x = \{\mathcal{B}, \mathbf{e}\}$ and returning a distance. With $h(x)/d_H$-structure-approximate-MPE, we define the problem of finding a structure approximation that differs in *at most* $h(x)$ variables from the most probable explanation $\mathit{optsol}(\mathcal{B}, \mathbf{e})$. With $E(h(x))/d_H$-structure-approximate-MPE we define the problem of finding a joint value assignment that has an *expected* Hamming distance $h(x)$ to $\mathit{optsol}(\mathcal{B}, \mathbf{e})$, i.e., a structure approximation is sought that differs *on average* in at most $h(x)$ variables from the MPE.

3.1 Computational complexity

In this section we will discuss the computational complexity of structure approximations of MPE. Note that a *random guess* of the values of variables would return a value assignment which gives an *expected* Hamming distance $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$, with c as the cardinality of the (unobserved) variables. In particular, when all unobserved variables are binary, we can expect to guess half of them correctly.

Corollary 1. *MPE is $E(h(x))/d_H$-structure approximable for $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$.*

We cannot expect to do better than chance: given that it is NP-hard to $\frac{n}{2} - \epsilon/d_H$-structure approximate 3SAT [6] and we can reduce 3SAT to MPE in polynomial time while preserving the structure of the certificates (by a simple variant of the proof used in [12, p.1457], which is omitted here for reasons of space), any polynomial-time $|\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon/d_H$-structure approximation algorithm for MPE could be used to find a $\frac{n}{2} - \epsilon/d_H$-structure approximation of any 3SAT instance in polynomial time.

Lemma 1. *MPE is $h(x)/d_H$-structure inapproximable for $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon$, unless* P = NP.

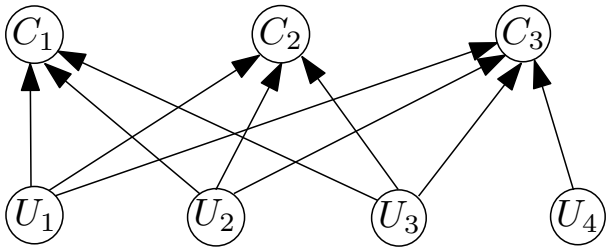

Fig. 2. Construction of $\mathcal{B}_{\phi_{\text{ex}}}$ from ϕ_{ex}

This result holds for binary variables with indegree at most three[3]. Here, we allow the approximation algorithm to select the $h(x)$ variables. If we are allowed to *designate* the variables for which the value is sought, then it is easy to see that we cannot have a polynomial-time structure approximation algorithm A for MPE, even for a single variable, unless $\mathsf{P} = \mathsf{NP}$, as we could use A consecutively for all $|\mathbf{M}|$ unobserved variables of $\mathcal{B}$ and thus obtain a polynomial-time exact algorithm for MPE; as MPE is NP-hard, the result follows as a corollary. However, we can prove a much stronger result for networks with three values per variable and indegree at most six: There cannot exist an algorithm that tells[4] us the value of an *arbitrary* single variable, unless $\mathsf{P} = \mathsf{NP}$:

Theorem 1. *No algorithm can calculate the value of one of the variables in the most probable explanation in polynomial time, unless* $\mathsf{P} = \mathsf{NP}$.

We will prove Theorem 1 with a reduction from 3SAT, defined as follows.

3-CNF SATISFIABILITY (3SAT)
Instance: A Boolean formula $\phi = (U, C)$ in 3-CNF form, with variables $U = u_1, \ldots, u_n$ and literals $C = c_1, \ldots, c_m$.
Question: Does there exist a truth assignment to the variables U such that all clauses C are satisfied?

As a running example, we will construct a network for the following (satisfiable) 3SAT instance [5]:

Example 1. $\phi_{\text{ex}} = (U, C)$, where $U = \{u_1, u_2, u_3, u_4\}$, and $C = \{(u_1 \vee u_2 \vee u_3), (\neg u_1 \vee \neg u_2 \vee u_3), (u_2 \vee \neg u_3 \vee u_4)\}$.

We construct a Bayesian network $\mathcal{B}_\phi$ from a 3SAT instance $\phi = (U, C)$ as follows. For each variable u_i in ϕ we add a *ternary* stochastic variable U_i in $\mathcal{B}_\phi$

[3] As each clause has three variables, the corresponding MPE instance has indegree at most three.

[4] Note that here we require that the algorithm not only returns a joint value assignment $cansol(x)$, but also tells us *which subset* of $cansol(x)$ matches $optsol(x)$.

with values $\{\text{TRUE}, \text{FALSE}, \#\}$ and uniform prior probability; the set of all U_i is denoted $\mathbf{U}$. For each clause c_j in ϕ we add a binary stochastic variable C_j in $\mathcal{B}_\phi$ with values TRUE and FALSE; the set of all C_j is denoted $\mathbf{C}$. C_j is to be conditioned on the variables $\mathbf{U_j} = \{U_j^1, U_j^2, U_j^3\}$ that correspond to the variables that occur in c_j, *and* (for $j > 1$) on the variables $\mathbf{U_{j-1}} = \{U_{j-1}^1, U_{j-1}^2, U_{j-1}^3\}$ that correspond to the variables that occur in c_{j-1}. To improve readability, we define the following shorthands for joint value assignments to $\mathbf{U_j}$ and $\mathbf{U_{j-1}}$: let $\mathbf{u}_\#$ denote a joint value assignment where *all* variables have the value #, and let $\mathbf{u_{TF}}$ denote a joint value assignment where *none* of the variables have the value #, i.e., all are TRUE or FALSE. For $C_j (j > 1)$ the following conditional probability distribution is defined.

$$\Pr(C_j = \text{TRUE} \mid \mathbf{U_j}, \mathbf{U_{j-1}}) = \begin{cases} 1 \text{ if } \mathbf{U_j} = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_j \text{ true,} \\ \quad \text{and } \mathbf{U_{j-1}} = \mathbf{u_{TF}} \\ \epsilon \text{ if } \mathbf{U_j} = \mathbf{u}_\# \text{ and } \mathbf{U_{j-1}} = \mathbf{u}_\# \\ 0 \text{ otherwise} \end{cases}$$

Here, ϵ is defined to be a sufficiently small (i.e., $\epsilon < \frac{1}{2^n}$), yet polynomial-time computable, value. Likewise, C_1 is defined as follows.

$$\Pr(C_1 = \text{TRUE} \mid \mathbf{U_1}) = \begin{cases} 1 \text{ if } \mathbf{U_1} = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_1 \text{ true} \\ \epsilon \text{ if } \mathbf{U_1} = \mathbf{u}_\# \\ 0 \text{ otherwise} \end{cases}$$

As an example of this construction, Figure 2 shows the network as constructed from ϕ_{ex}. We set the evidence variables $\mathbf{E} = \mathbf{C}$ with $\mathbf{e} = \wedge_{j=1}^{m} C_j = \text{TRUE}$. We claim that ϕ is satisfiable if and only if *none* of the variables in the most probable joint value assignment $\mathbf{u}$ to $\mathbf{U}$ has the value #, and unsatisfiable if and only if *all* of the variables in $\mathbf{u}$ have the value #. Thus, if an approximation algorithm tells us the value of *any* variable of the most probable explanation of $\mathcal{B}$, we can use that algorithm to solve the corresponding 3SAT instance in polynomial-time.

Proof (of Theorem 1). Assume there exists a polynomial-time structure approximation algorithm A that, when given an MPE instance, returns for one of the variables in the explanation set M a value that corresponds to the value of that variable in the most probable explanation. We will show that A can be used to decide 3SAT in polynomial time; hence, from the existence of such an algorithm it would follow that $\mathsf{P} = \mathsf{NP}$. Let ϕ be an arbitrary instance of 3SAT and let $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$ be the MPE instance as constructed above. Note that we can construct $\mathcal{B}_\phi$ from ϕ in polynomial time, as every literal and clause in ϕ corresponds to a single variable in $\mathcal{B}_\phi$ and the size of the conditional probability tables of each variable is bounded by a constant.

Let $\mathbf{u}$ be a joint value assignment to the variables of $\mathbf{U}$ of $\mathcal{B}_\phi$. We will distinguish between three possible scenarios:

1. $\mathbf{u} \in \{\#\}^n$, i.e., *all* variables are set to #
2. $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$, i.e., *none* of the variables are set to #

3. $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}, \#\}^n$, $\mathbf{u} \notin \{\#\}^n$, and $\mathbf{u} \notin \{\text{TRUE}, \text{FALSE}\}^n$

Note that in case 3) $\Pr(\mathbf{u}, \mathbf{e}) = 0$ due to the constraints in the joint probability distributions of C_j. In case 2), if $\mathbf{u}$ does *not* satisfy ϕ, then also $\Pr(\mathbf{u}, \mathbf{e}) = 0$. If on the other hand $\mathbf{u}$ *does* satisfy ϕ, then the probability $\Pr(\mathbf{u}, \mathbf{e})$ equals $\frac{1}{N_{sat}(1+\epsilon)}$, where $1 \leq N_{sat} \leq 2^n$ denotes the number of satisfying truth assignments to ϕ. In case 1), if ϕ is satisfiable, then $\Pr(\mathbf{u}, \mathbf{e}) = \frac{\epsilon}{1+\epsilon}$; as ϵ was chosen to be strictly less than $\frac{1}{2^n}$, this probability is lower than the probability of any satisfying joint value assignment. However, when ϕ is not satisfiable, then $\Pr(\mathbf{u}, \mathbf{e}) = 1$.

Thus, the most probable explanation for evidence $\mathbf{e} = \wedge_{j=1}^{m} C_j = \text{TRUE}$ is either $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$ if ϕ is satisfiable, or $\mathbf{u} \in \{\#\}^n$ if ϕ is not satisfiable. Now assume that, when given $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$ as input, A outputs the value assignment of one of the unobserved variables in $\mathcal{B}_\phi$, that correspond to the value in the most probable explanation of $\mathcal{B}_\phi$. In case A outputs TRUE or FALSE, ϕ is satisfiable; in case A outputs #, ϕ is not satisfiable. Hence, we can use A to solve 3SAT in polynomial time, concluding the proof.

3.2 Ordered variables

We saw in the previous section that it is NP-hard to structure-approximate even a single variable of the most probable explanation in a Bayesian network. However, we assumed that the values of the variables in the network were *unordered*. In this section we assume a particular order on the values and investigate the consequences for the computational complexity of structure approximation.

Typically, in a Bayesian network some variables might have a 'natural' ordering, like a variable HEIGHT with values TALL, NORMAL and SMALL; these values are ordered $\text{SMALL} \preceq \text{NORMAL} \preceq \text{TALL}$. Other variables, like BLOODTYPE or ETHNICGROUP lack such an ordering. When a variable is ordered, it makes sense to redefine the distance measure: when HEIGHT is assigned the value TALL in the most probable explanation, NORMAL would be a better approximation than SMALL.

In the remainder we assume that all variables are ordered, and we introduce a *partial ordered lattice* [20] and a corresponding *lattice distance function*. The lattice includes all joint value assignments to the observable variables in the network and it captures the partial order between the assignments. The bottom of the lattice encodes the joint value assignment $\mathbf{m}$ such that $\mathbf{m} \preceq \mathbf{m}'$ for all $\mathbf{m}' \in \Omega(\mathbf{M})$. Likewise, the top of the lattice encodes the joint value assignment $\mathbf{m}''$ such that $\mathbf{m}' \preceq \mathbf{m}''$ for all $\mathbf{m}' \in \Omega(\mathbf{M})$. In general, a lattice element $L(\mathbf{m})$ encoding a joint value assignment $\mathbf{m}$ precedes another lattice element $L(\mathbf{m}')$ if and only if $\mathbf{m} \preceq \mathbf{m}'$. In Figure 3 an example (from [20]) is shown for two ternary variables X and Y.

A natural distance function comparing two joint value assignments $\mathbf{m}$ and $\mathbf{m}'$ would be the *distance in the lattice* between these assignments, i.e., the length of the shortest path from $L(\mathbf{m})$ to $L(\mathbf{m}')$. For example, the distance between x_2y_1 and x_1y_3 would be three. Note that this distance function, denoted by d_L, is a metric as the properties of Section 2.2 also hold for d_L. Using this distance

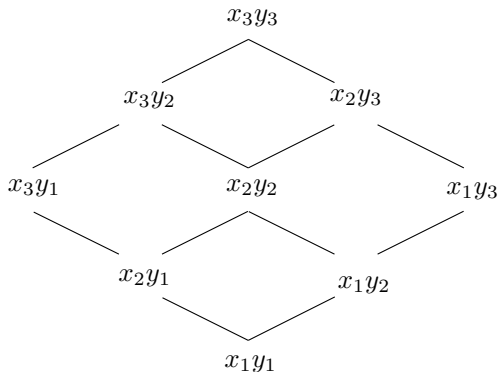

Fig. 3. A lattice describing the partial order of the joint value assignments to the variables X and Y

function, we can find a trivial *guaranteed* $h(x)/d_L$-structure approximation with ordering for $h(x) = |\mathbf{M}| \cdot \lfloor \frac{c}{2} \rfloor$, rather than the *expected* $E(h(x)) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$ without ordering, by always picking the 'middle' value in the order. We can, however, not expect to do better than $h(x) = |\mathbf{M}|$ for $c \geq 5$, unless $\mathsf{P} = \mathsf{NP}$:

Theorem 2. *MPE is $h(x)/d_L$-structure inapproximable for $h(x) = |\mathbf{M}| - 1$, unless* $\mathsf{P} = \mathsf{NP}$.

Proof. Similar as in the proof of Theorem 1, and using the same construction, we show that the existence of a polynomial-time algorithm A that can $h(x)/d_L$-structure-approximate MPE for $h(x) = |\mathbf{M}| - 1$ implies that we can decide 3SAT in polynomial time. We augment the construction used to prove Theorem 1 as follows: let all variables U_i have *five* values $\Omega(U_i) = \{\text{FALSE}, \text{TRUE}, \#, d_1, d_2\}$ in which d_1 and d_2 act as dummy variables. U_i is uniformly distributed, and the order of $\Omega(U_i)$ is $\text{FALSE} \preceq d_1 \preceq \# \preceq d_2 \preceq \text{TRUE}$. The conditional probability distribution of C_j is similar as in Theorem 1; in particular, any joint value assignment $\mathbf{U}$ that includes a dummy variable has probability $\Pr(C_1 = \text{TRUE} \mid \mathbf{U}) = 0$. We claim that, for any $h(x)/d_L$-structure approximation with $h(x) \leq |\mathbf{M}| - 1$, the majority of the variables that contain non-dummy values can be used to decide satisfiability of ϕ: if the (strict) majority of these variables has TRUE or FALSE as value, then the instance is satisfiable, otherwise the instance is unsatisfiable.

Observe that an approximation with $h(x) = |\mathbf{M}| - 1$ has at least *one* 'correct' variable, as any deviation from the MPE would increase $h(x)$ by at least one, i.e., every variable that has a value that is not equal to the MPE contributes a distance of 1 to $h(x)$. In particular, when one of the variables is correctly labeled with either # (for an unsatisfying instance) or TRUE or FALSE (for a satisfying instance), and the other variables have dummy values that are closest to the MPE value of that variable (i.e., d_1 for FALSE, d_2 for TRUE, and either d_1 or

d_2 for #), then $h(x) = |\mathbf{M}| - 1$; clearly here a majority of the (non-dummy) variables correctly reflects the satisfiability of the instance.

Now we show that this property holds for every alteration to this joint value assignment that maintains that $h(x) = |\mathbf{M}| - 1$. We will demonstrate the case that ϕ is satisfiable; for unsatisfiable ϕ, the proof goes analogously.

- If we replace a dummy value with a # value, then $h(x)$ increases by one. We must also change another dummy value to TRUE or FALSE (whichever is closest) to maintain that $h(x) = |\mathbf{M}| - 1$, so still the majority of non-dummy variables has as value TRUE or FALSE.
- If we replace a TRUE or FALSE value to a # value, then $h(x)$ increases by two, and so two dummy variables need to be changed into TRUE or FALSE.

Thus, if A returns a $h(x)/d_L$-structure approximation with $h(x) \leq |\mathbf{M}| - 1$, then we can use the output to decide 3SAT: count the number TRUE or FALSE values and the number of #-values. If the first number is higher than the second, answer *yes*, else answer *no*. As A runs in polynomial time, this algorithm can decide 3SAT in polynomial time, hence $\mathsf{P} = \mathsf{NP}$.

4 Conclusion

In this paper we discussed structure approximations of MPE. In general, we cannot do better than just randomly guess the joint value assignment: we then would on average expect to guess $\frac{1}{c}$ of the variables correctly, where c is the cardinality of the variables. As it is NP-hard to determine the value of more than $\frac{1}{c}$ of the variables in the MPE, there is little room for improvement. We hypothesize (but could not prove) that it is even NP-hard to get an *expected* structure approximation that is strictly better than $|\mathbf{M}| - \frac{|\mathbf{M}|}{c}$.

Furthermore, we showed that it is NP-hard in general to obtain an approximation that determines even a *single* variable in the MPE. So, without information on the ordering of the values or restrictions on the network structure or probability distribution, if we want information on the structure of the MPE (in polynomial time), there are little alternatives than to compute it exactly.

However, if we do have information on the ordering of the values, we can do a bit better[5] than that. We showed that the simple strategy 'always stay in the middle' *guarantees* a $h(x)/d_L$-structure approximation for $h(x) = |\mathbf{M}| \cdot \lfloor \frac{c}{2} \rfloor$ in the worst case, which is better than the expected value if we would randomly guess the values. We showed that it is NP-hard to $h(x)/d_L$-structure approximate MPE for $h(x) = |\mathbf{M}| - 1$ and $c \geq 5$.

The gap between these two results (for $c = 5, h(x) = 2 \cdot |\mathbf{M}|$) might leave some room for improvement. There may be constrained situations where solving MPE exactly remains intractable, whereas a good structure approximation might be

[5] As one reviewer carefully pointed out, the Hamming and edit distances are not quite comparable for $c > 2$ as the edit distance will be on average larger with larger c, while the Hamming distance remains 1 whenever a mismatch occurs.

found in polynomial time. One suggestion, that we leave for future work, is to investigate whether it could help to use *monotonicity properties* in the network to get a better structure approximation; the NP-hardness proofs in this paper critically depend on non-monotone relations between the clause-nodes and the literal-nodes in the network. However, note that even when the hypothesis space is monotone in the evidence, obtaining evidence need not tell us anything about the most probable hypothesis. As an example, in the following conditional dependencies for H_1 and H_2, the most probable hypothesis given evidence e differs, even though both H_1 and H_2 are monotone in E:

$$\Pr(H_1 = \text{TRUE} \mid E) = \begin{cases} 0.2 \text{ if } E = \text{TRUE} \\ 0.1 \text{ if } E = \text{FALSE} \end{cases}$$

$$\Pr(H_2 = \text{TRUE} \mid E) = \begin{cases} 0.7 \text{ if } E = \text{TRUE} \\ 0.6 \text{ if } E = \text{FALSE} \end{cases}$$

Note, that $\text{argmax}_{H_1} \Pr(H_1, e = \text{TRUE}) = \text{FALSE}$, while $\text{argmax}_{H_2} \Pr(H_2, e = \text{TRUE}) = \text{FALSE}$.

Acknowledgments

The author wishes to thank Iris van Rooij, Linda van der Gaag, and Todd Wareham for helpful discussions on this topic, and the anonymous reviewers for their constructive comments. A previous version of this paper appeared in the 24th Benelux Conference on Artificial Intelligence [13].

References

1. A. M. Abdelbar and S. M. Hedetniemi. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38, 1998.
2. C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.
3. H. L. Bodlaender, F. van den Eijkhof, and L. C. van der Gaag. On the complexity of the MPA problem in probabilistic networks. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 675–679, 2002.
4. N. Chater and M. Oaksford. Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, 3:5765, 1999.
5. G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990.
6. U. Feige, M. Langberg, and K. Nissim. On the hardness of approximating NP witnesses. In K. Jansen and S. Khuller, editors, *Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, volume 1913 of *LNCS*, pages 120–131, 2000.
7. M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., San Francisco, CA, 1979.

8. J. Gemela. Financial analysis using Bayesian networks. *Applied Stochastic Models in Business and Industry*, 17:57–67, 2001.
9. M. Hamilton, M. Müller, I. van Rooij, and T. Wareham. Approximating solution structure. In E. Demaine, G.Z. Gutin, D. Marx, and U. Stege, editors, *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs*, number 07281 in Dagstuhl Seminar Proceedings, 2007.
10. F. V. Jensen and T. D Nielsen. *Bayesian Networks and Decision Graphs*. Springer Verlag, New York, NY, second edition, 2007.
11. R. Kumar and D. Sivakumar. Proofs, codes, and polynomial-time reducibilities. In *Proceedings of the Fourteenth Annual Conference on Computational Complexity*, pages 46–53. IEEE Computer Society, 1999.
12. J. Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52(9):1452 – 1469, 2011.
13. J. Kwisthout. Structure approximation of most probable explanations in Bayesian networks. In J.W.H.M. Uiterwijk, N. Roos, and M.H.M. Winands, editors, *Proceedings of the 24th Benelux Conference on Artificial Intelligence (BNAIC'12)*, pages 131–138, 2012.
14. C. Lacave and F. J. Díez. A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, 17(2):107–127, 2002.
15. P. J. F. Lucas, N. de Bruijn, K. Schurink, and A. Hoepelman. A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 3:251–279, 2000.
16. E. Millgram. Coherence: The price of the ticket. *Journal of Philosophy*, 97:82–93, 2000.
17. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Palo Alto, CA, 1988.
18. S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, 1994.
19. P. J. Sticha, D. M. Buede, and R. L. Rees. Bayesian model of the effect of personality in predicting decisionmaker behavior. In L. C. Van der Gaag and R. Almond, editors, Proceedings of the Fourth Bayesian Modelling Applications Workshop, 2006.
20. L. C. van der Gaag, S. Renooij, and P. L. Geenen. Lattices for studying monotonicity of Bayesian networks. In M. Studený and J. Vomlel, editors, *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pages 99–106, 2006.
21. L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, and B. G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.
22. I. van Rooij, J. Kwisthout, M. Blokpoel, J. Szymanik, T. Wareham, and I. Toni. Communicating intentions: Computationally easy or difficult? *Frontiers in Human Neuroscience*, 5(52):1–18, 2011.
23. I. van Rooij and T. Wareham. Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, 56(4):232–247, 2012.