Twelfth Scandinavian Conference on Artificial Intelligence
M. Jaeger et al. (Eds.)
IOS Press, 2013

doi:10.3233/978-1-61499-330-8-275

Probabilistic Inference in BN2T Models by Weighted Model Counting

Jiří VOMLEL [a,b,1] and Petr TICHAVSKÝ [a]

[a] *Institute of Information Theory and Automation of the AS CR, Prague, The Czech Republic*

[b] *Faculty of Management, University of Economics, Prague, Jindřichův Hradec, The Czech Republic*

Abstract. Exact inference in Bayesian networks with nodes having a large parent set is not tractable using standard techniques as are the junction tree method or the variable elimination. However, in many applications, the conditional probability tables of these nodes have certain local structure than can be exploited to make the exact inference tractable. In this paper we combine the CP tensor decomposition of probability tables with probabilistic inference using weighted model counting. The motivation for this combination is to exploit not only the local structure of some conditional probability tables but also other structural information potentialy present in the Baysian network, like determinism or context specific independence. We illustrate the proposed combination on BN2T networks – two-layered Bayesian networks with conditional probability tables representing noisy threshold models.

Keywords. Bayesian Networks, Independence of Causal Influence, Noisy Threshold Models, Probabilistic Inference, Weighted Model Counting, Arithmetic Circuits

Introduction

Bayesian networks [1,2] are a popular model for reasoning under uncertainty. Computationally efficient probabilistic inference is possible even in models with hundreds of variables using techniques [3,4] that exploit the conditional independence relations between modeled variables encoded by an acyclic directed graph. Unfortunately, if a node in the graph has a very large parent set the exact inference using standard techniques as are the junction tree method or the variable elimination is not tractable. On the other hand, in many real application of Bayesian networks the conditional probability tables (CPTs) have certain local structure that can be exploited to make the exact inference tractable. A class of CPTs with local structure are models with independence of causal influence (ICI models) [5] – a subclass of so called canonical models [6].

Diverse inference methods that exploit the local structure of CPTs were proposed. In this paper we combine one such method – CP tensor decomposition [7,8,9] – with weighted model counting applied to probabilistic inference [10,11]. In [10] it was ex-

[1] Corresponding Author: Jiří Vomlel, Institute of Information Theory and Automation, Pod vodárenskou věží 4, Prague 8, 182 08, Czech Republic; E-mail: vomlel@utia.cas.cz.
This work was supported by the Czech Science Foundation through the projects 13-20012S and 102/09/1278.

perimentaly shown that a significant computational gain can be achieved if the CPTs are transformed before a weighted model counter is applied. We illustrate our approach by use of a special class of Bayesian networks that is well motivated by applications. This class is a generalization of BN2O networks, which are two-layered Bayesian networks with conditional probability tables representing noisy-or models. A popular example of this class is the decision theoretic version of the Quick Medical Reference model (QMR-DT) [12]. We generalize BN2O networks to BN2T networks by replacing the noisy-or by its generalization, the noisy threshold model. This allows modeling of a synergic effect of causes, e.g., if for a symptom to be observed as positive more than one cause (or defect) needs to be present.

The paper is organized as follows. In Section 1 we introduce the necessary notation. In Section 2 we present the Bayesian network factorization formula, which we will utilize in consequent sections. Models of independence of causal influence are introduced in Section 3, where also the CP tensor decomposition of these models is described. The core of the paper is Section 4, where the CP tensor decomposition of CPTs of BN2T networks is combined with weighted model counting. In Section 5 the contribution of the paper is summarized our future research plans are presented.

1. Preliminaries

Let $V = \{1,\ldots,n\}, n \in \mathbb{N}$. For $i \in V$ define variables X_i taking states x_i from a finite set $\mathbb{X}_i$. Further let $X_A, A \subseteq V$ denote a multidimensional variable $(X_j)_{j\in A}$, let $x_A = (x_j)_{j\in A}$ denote a configuration of values of variable X_A, and let $\mathbb{X}_A = \times_{j\in A}\mathbb{X}_i$ denote the set of all configurations of values of variable X_A. We will use abbreviations $X = X_V$, $x = x_V$, $\mathbb{X} = \mathbb{X}_V$, $X_i = X_{\{i\}}$, $x_i = x_{\{i\}}$, and $\mathbb{X}_i = \mathbb{X}_{\{i\}}$.

Definition 1. A *table* $\psi_{X_A}, A \subseteq V$ is a function $\psi_{X_A} : \mathbb{X}_A \to \mathbb{R}$ viewed as an $|A|$-dimensional array (also called tensor of order $|A|$), where coordinates in each dimension are given by the values x_i of variable X_i corresponding to that dimension. For a value of function ψ_{X_A} at a point x_A we will often write $\psi(x_A)$ instead of $\psi_{X_A}(x_A)$.

Definition 2. A *conditional probability table* (CPT) $P_{X_A|X_B}$ is a table $\psi_{X_{A\cup B}}, A, B \subseteq V, A \cap B = \emptyset$ such that it holds for all $(x_A, x_B) \in \mathbb{X}_A \times \mathbb{X}_B$ that $0 \leq \psi(x_A, x_B) \leq 1$ and $\sum_{x_A} \psi(x_A, x_B) = 1$. If it is clear from context then for a particular combination of values x_A, x_B of $P_{X_A|X_B}$ we write $P(x_A|x_B)$.

Remark. If $B = \emptyset$ then for $P_{X_A|X_B}$ we use abbreviation P_{X_A} and call it *probability table*.

CPTs can be viewed as multidimensional arrays (tensors), that can be visualized using nested matrices. In this paper we will follow the convention that the first variable defines the most outer row coordinate, the second variable the most outer column coordinate, etc. See Example 1.

Example 1. The conditional probability table $P_{Y|X_1,X_2,X_3}$ can be viewed as a four-dimensional array (tensor). Let $P_{Y|X_1,X_2,X_3}$ represent logical or, i.e., let

$$P(y|x_1,x_2,x_3) = (y \iff (x_1 \vee x_2 \vee x_3)), \tag{1}$$

where logical values are 0 and 1. We can visualize the corresponding array (tensor) using nested matrices with successive dimensions alternating between rows and columns as:

$$P_{Y|X_1,X_2,X_3} = \begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{pmatrix} .$$

Definition 3 (Restriction)**.** In a table ψ_{X_A} we may fix values $x_B \in \mathbb{X}_B$ of some variables $X_B, \emptyset \neq B \subseteq A$. Then we get a new table $\psi_{X_C, X_B = x_B}$, where $C = A \setminus B$. It is defined for all values x_C as $\psi_{X_C, X_B = x_B}(x_C) = \psi_{X_A}(x_B, x_C)$.

Example 2. Let $P_{Y|X_1,X_2,X_3}$ represent logical or as it was defined in Example 1. Then

$$P_{Y=1|X_1,X_2,X_3} = \begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} .$$

Definition 4 (Multiplication)**.** Let ψ_{X_A} and φ_{X_B} be two tables such that $A, B \subseteq V$. The product $\xi_{X_{A \cup B}}$ of ψ_{X_A} and φ_{X_B} is defined as $\xi(x_A, x_B) = \psi(x_A) \cdot \varphi(x_B)$.

Example 3. In this example we multiply three tables $P_{Y|X_1,X_2}$, P_{X_1}, and P_{X_2}

$$P_{Y|X_1,X_2} = \begin{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix}, \quad P_{X_1} = \begin{pmatrix} p_1 \\ 1 - p_1 \end{pmatrix}, \text{ and } P_{X_2} = \begin{pmatrix} p_2 \\ 1 - p_2 \end{pmatrix} .$$

The product $\psi_{Y,X_1,X_2} = P_{Y|X_1,X_2} \cdot P_{X_1} \cdot P_{X_2}$ is

$$\psi_{Y,X_1,X_2} = \begin{pmatrix} \begin{pmatrix} p_1 \cdot p_2 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ p_1 \cdot (1 - p_2) \end{pmatrix} & \begin{pmatrix} (1 - p_1) \cdot p_2 \\ (1 - p_1) \cdot (1 - p_2) \end{pmatrix} \end{pmatrix} .$$

Definition 5 (Marginalization)**.** Let ψ_{X_A} be a table and $\emptyset \neq B \subset A$. ψ_{X_B} is a marginal table of ψ_{X_A} if it holds that: $\psi_{X_B}(x_B) = \sum_{x_C} \psi_{X_A}(x_B, x_C)$ where $C = A \setminus B$.

Example 4. Consider table ψ_{Y,X_1,X_2} from Example 3. Table ψ_{X_1,X_2} is its marginal:

$$\psi_{X_1,X_2} = \begin{pmatrix} p_1 \cdot p_2 & (1 - p_1) \cdot p_2 \\ p_1 \cdot (1 - p_2) & (1 - p_1) \cdot (1 - p_2) \end{pmatrix} . \tag{2}$$

2. Bayesian networks

Bayesian networks [1,13,2] describe probabilistic relations between random variables $X_1, \ldots, X_n$. The structure of a Bayesian network is defined by an acyclic directed graph

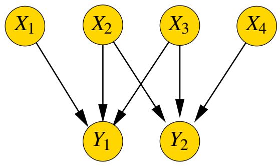

Figure 1. Structural part of the Bayesian network from Example 5.

$G = (V, E)$, where E is the set of directed edges, i.e., $E \subseteq V \times V$. The joint probability of a Bayesian network is defined for all configurations $x = (x_1, \ldots, x_n)$ of the multidimensional variable $X = (X_1, \ldots, X_n)$ as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | x_{pa(i)}) \quad , \tag{3}$$

where $pa(i)$ denotes the set of parents[2] of node i in the graph $G = (V, E)$, i.e. $pa(i) = \{j, (i \rightarrow j) \in E\}$. The factorization defined by formula (3) allows efficient computations of probabilistic queries $P(X_i|e), e = \{X_j = x_j, j \in A\}, A \subset V$ for all $i \in V \setminus A$. This allows applications of Bayesian networks in domains with hundreds of variables, where a naive computations with the full joint probability table would not be tractable.

Remark. To differentiate between different groups of variables we will denote variables also by different letters than X, for example by Y or Y', etc.

Example 5. In Figure 1 we give an example of an acyclic directed graph that defines the structural part of a Bayesian network. The joint probability of this Bayesian network is defined for all configurations $(x_1, x_2, x_3, x_4, y_1, y_2)$ as

$$P(x_1, \ldots, x_4, y_1, y_2) = P(y_1|x_1, x_2, x_3) \cdot P(y_2|x_2, x_3, x_4) \cdot P(x_1) \cdot P(x_2) \cdot P(x_3) \cdot P(x_4) \quad .$$

3. Models of Independence of Causal Influence

Unfortunately, in some application, the treewidth is large, often because some variables $X_i, i \in V$ have a large parent set, e.g., $|pa(i)| > 100$ and the exact inference with the standard junction tree method is not tractable. Luckily, in many real application of Bayesian networks the conditional probability tables $P(X_i|X_{pa(i)})$ have a certain local structure that can be exploited to make the exact inference tractable. A class of CPTs with local structure are models with independence of causal influence (ICI models) [5], which is a subclass of so called canonical models [6]. In each ICI model it is possible to make graphically explicit the deterministic and probabilistic parts using auxiliary variables $X'_i, i = 1, \ldots, k$. On the left hand side of Figure 2 a structure of an ICI model is presented.

A most popular example of an ICI model is the noisy-or model defined as

$$P_{Y|X_1,\ldots,X_k}(0|x_1, \ldots, x_k) = \prod_{i=1}^{k} (p_i)^{x_i} \quad , \tag{4}$$

[2]For simplicity, we will also use $pa(X_i)$ to refer to variables corresponding to nodes that are parents of the node i corresponding to variable X_i.

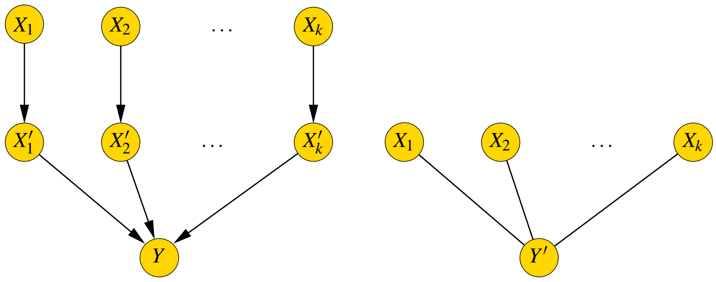

Figure 2. Structure of an ICI model on the left hand side and the graph of the CP decomposition of an ICI model with an evidence $Y = y$ on the right hand side .

where parameters $p_i, i = 1, \dots, k$ satisfying $0 \le p_i < 1$ are called inhibitory probabilities.

Example 6 (Noisy-or model for $k = 3$)**.** The CPTs are defined as follows:

$$P_{X_i'|X_i} = \begin{pmatrix} 1 & p_i \\ 0 & 1-p_i \end{pmatrix} \text{ for } i = 1, \dots, k, \ P_{Y|X_1',X_2',X_3'} = \begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \end{pmatrix} , \quad (5)$$

where parameters p_i represent probabilities that the positive influence of $X_i = 1$ on Y is inhibited with probability p_i.

A generalization of the noisy-or model is the noisy threshold which takes value $Y = 1$ if at least ℓ out of k parent variables $X_1', \dots, X_k'$ are true ($0 < \ell \le k$). Let $K = \{1, \dots, k\}$. The CPT of the noisy threshold model is defined by

$$P_{Y|X_1,\dots,X_k}(0|x_1,\dots,x_k) = \sum_{j=0}^{\ell-1} \sum_{I \subseteq K, |I|=k-j} \prod_{i \in I} (p_i)^{x_i} \prod_{j \in K \setminus I} (1-(p_j)^{x_j}) , \quad (6)$$

Note that for $\ell = 1$ the noisy threshold model corresponds to the noisy-or model. The threshold functions appear, for example, in medical applications [14,15,16].

Example 7 (Noisy threshold for $k = 3, \ell = 2$)**.** In a noisy-threshold model with $k = 3$ the CPTs $P_{X_i'|X_i}, i = 1, \dots, k$ are defined as for noisy-or in Example 6 and $P_{Y|X_1',X_2',X_3'}$ is defined as follows:

$$P_{Y|X_1',X_2',X_3'} = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \end{pmatrix} . \quad (7)$$

Diverse inference methods that exploit the local structure of CPTs were proposed. Even a brief review of these methods would go beyond the scope of this paper. Instead we refer to other papers [17, Section 3] or [7, page 753], where methods exploiting local structure in probabilistic inference are listed together with references to original papers.

In this paper we build on one of the proposed methods – CP tensor decomposition (previously called tensor rank-one decomposition) [7,8]. The CP tensor decomposition is a generalization of the Díez and Galán's decomposition of noisy-max originally proposed in [9]. In [7] it was generalized to some other canonical models. In [8] a CP tensor decomposition of noisy threshold models was proposed. The basic idea of the CP tensor decomposition is by introducing an auxiliary variable Y' rewrite the CPT as a marginal of a product of two-dimensional tables. This is always possible if we allow sufficient number of states Y'. For a given CPT the goal is to find a decomposition with minimal number of states $|\mathbb{Y}'|$ of Y'. The minimal value of $|\mathbb{Y}'|$ is called the *rank* of table (array, tensor) $P_{Y|X_1,\ldots,X_k}$. In this paper we deal with decompositions of CPTs where Y has its state y observed – either $y=0$ or $y=1$.

$$P_{Y=y|X_1,\ldots,X_k} = \sum_{Y'} \prod_{i=1}^{k} \psi_{X_i,Y'} \quad . \tag{8}$$

This decomposition can be visualized using an undirected graph, see right hand side of Figure 2. In [8] an algorithm for CP-decomposition of noisy-threshold was presented that requires $|\mathbb{Y}'| = |pa(Y)|$.

Example 8 (Decomposition of threshold for $k=3, \ell=2$ and $Y=1$)**.** Consider the deterministic part $P_{Y|X'_1,X'_2,X'_3}$ of a noisy-threshold model from Example 7 and assume that $Y=1$. Then using the algorithm described in [8] we can get the following CP tensor decomposition:

$$P_{Y=1|X'_1,X'_2,X'_3} = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} = \sum_{Y'} \prod_{i=1}^{3} \psi_{X'_i,Y'}, \tag{9}$$

$$\psi_{X'_i,Y'} = \begin{pmatrix} \sqrt[3]{\frac{1}{6}} & \sqrt[3]{\frac{1}{3}} & -\sqrt[3]{\frac{1}{2}} \\ 2\sqrt[3]{\frac{1}{6}} & -\sqrt[3]{\frac{1}{3}} & 0 \end{pmatrix} \quad \text{for } i = 1,2,3 \quad . \tag{10}$$

The CP tensor decomposition can be applied to Bayesian networks having defined their structure by arbitrary acyclic directed graphs. However, in this paper we decided to deal only with Bayesian networks that have the structure of a bipartite graph with all edges directed from one part (the top level) towards the other (the bottom level) and all conditional probability tables represent noisy threshold models. These networks are a generalization of BN2O networks, generalizing noisy-or to noisy threshold models. We will refer to them as to BN2T networks. See Figure 1 for an example of the graph of a BN2T network. Furthermore, we assume that only the nodes from the bottom level can be observed[3]. This allows us to include in the model only CPTs of bottom level nodes that were observed since the unobserved nodes from the bottom level are barren nodes [13]. This observation together with (3) implies that for an evidence $e = (y_1, \ldots, y_m)$ it holds that

[3]In medical applications this corresponds to observing symptoms, results of lab tests, etc. Diseases are not directly observable.

$$P(x_1,\ldots,x_n,e) = \prod_{i=1}^{n} P(x_i) \prod_{j=1}^{m} P(y_j|x_{pa(Y_j)}) \ , \tag{11}$$

where $X_i, i = 1,\ldots,n$ are the nodes in the top level and $Y_j, j = 1,\ldots,m$ are observed nodes in the bottom level. It follows by the Bayes rule that

$$P(x_1,\ldots,x_n|e) = \frac{1}{P(e)} \prod_{i=1}^{n} P(x_i) \prod_{j=1}^{m} P(y|x_{pa(Y_j)}) \ , \tag{12}$$

$$P(e) = \sum_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,e) \ . \tag{13}$$

We can substitute the CP tensor decomposition of CPTs $P_{Y_j=y_j|pa(Y_j)}, j = 1,\ldots,m$ as defined in formula (8) into formulas (11) and (13) and get

$$P(e) = \sum_{x_1,\ldots,x_n,y'_1,\ldots,y'_m} \left(\prod_{i=1}^{n} P(x_i)\right) \left(\prod_{j=1}^{m} \prod_{X_i \in pa(Y_j)} \psi_{X_i,Y'_j}(x_i,y'_j)\right) \ . \tag{14}$$

It is important to note that contrary to formula (11) the tables in formula (14) are defined for at most two variables. Typically, this allows substantially more efficient inference. The lower the number of states of auxiliary variables $Y'_1,\ldots,Y'_m$ the lower is the computational complexity. In [8] the computational complexity of the junction tree method applied to full CPT tables and to the tables after CP tensor decomposition was compared. Achieved savings were several orders of magnitudes for larger networks. In this paper we will use a completely different approach to computing the value of $P(e)$ based on weighted model counting of logical models [18,10,11].

4. Probabilistic Inference by Weighted Model Counting

It was shown that if Bayesian networks exhibit a lot of determinism or context specific independence the weighted model counting (WMC) represents an efficient method for probabilistic inference [11]. The basic idea of WMC is to encode a Bayesian network using a conjunctive normal form (CNF), associate weights to literals according to the CPTs of the Bayesian network, and than compute the probability of evidence as the sum of weights of all logical models consistent with that evidence. The weight of a logical model is the product of weights of all literals. Efficient WMC solvers exploiting several advanced techniques such as clause learning, component caching, etc can be used. An example of a successful WMC solver is Cachet [19].

Next we follow Chavira and Darwiche's approach [18] and show how a BN2T can be encoded as a CNF. This is similar to the approach of Wei Li et al. [17], who used the encoding of Sang et al. [10] to encode Bayesian networks with noisy max models.

- For each state $x \in \mathbb{X}_i$ of a BN2T variable X_i a logical variable $\lambda^x_{X_i}$ is created.
- For each state $y \in \mathbb{Y}'_j$ of a BN2T variable Y'_j a logical variable $\lambda^y_{Y'_j}$ is created.
- For each variable X_i following clauses that ensure the states of X_i are mutually exclusive take part in the CNF:

$$\bigvee_{x \in \mathbb{X}_i} \lambda^x_{X_i}, \ \text{and} \ (\neg\lambda^a_{X_i} \vee \neg\lambda^b_{X_i}) \ \text{for all } a,b \in \mathbb{X}_i, a < b \ . \tag{15}$$

- For each variable Y'_j following clauses that ensure that the states of Y' are mutually exclusive take part in the CNF:

$$\bigvee_{y \in \mathbb{Y}'_j} \lambda^{y}_{Y'_j} \text{ and } (\neg\lambda^{a}_{Y'_j} \vee \neg\lambda^{b}_{Y'_j}) \text{ for all } a,b \in \mathbb{Y}'_j, a < b \ . \tag{16}$$

- For each value of each table in formula (14) a logical variable is created. Variables $\theta^{x}_{X_i}, x = 0,1$ for each table $P_{X_i}, i = 1,\ldots,m$ and $\theta^{x,y}_{X_i,Y'_j}, x = 0,1, y = 1,\ldots,|pa(Y_j)|$ for each table ψ_{X_i,Y'_j}.
- For each logical variable $\theta^{x}_{X_i}, i = 1,\ldots,n$ and for each state $x \in \mathbb{X}$ two clauses are included in the CNF:

$$(\neg\theta^{x}_{X_i} \vee \lambda^{x}_{X_i}) \text{ and } (\theta^{x}_{X_i} \vee \neg\lambda^{x}_{X_i}) \ . \tag{17}$$

  They ensure that $\theta^{x}_{X_i}$ is true if and only if $\lambda^{x}_{X_i}$ is true.
- For each logical variable $\theta^{x,y}_{X_i,Y'_j}$ three clauses are included in the CNF:

$$(\theta^{x,y}_{X_i,Y'_j} \vee \neg\lambda^{x}_{X_i} \vee \neg\lambda^{y}_{Y'_j}) \text{ and } (\neg\theta^{x,y}_{X_i,Y'_j} \vee \lambda^{x}_{X_i}) \text{ and } (\neg\theta^{x,y}_{X_i,Y'_j} \vee \lambda^{y}_{Y'_j}) \ . \tag{18}$$

  They ensure that $\theta^{x,y}_{X_i,Y'_j}$ is true if and only if $\lambda^{x}_{X_i}$ and $\lambda^{y}_{Y'_j}$ is true.
- The weights of all positive literals are defined for all $x \in \mathbb{X}_i$ and $y \in \mathbb{Y}'_j$ as:

$$w(\lambda^{x}_{X_i}) \ = \ 1 \text{ and } w(\lambda^{y}_{Y'_j}) \ = \ 1 \tag{19}$$

$$w(\theta^{x}_{X_i}) \ = \ P_{X_i}(x) \text{ and } w(\theta^{x,y}_{X_i,Y'_j}) \ = \ \psi_{X_i,Y'_j}(x,y) \ . \tag{20}$$

- The weights of all negative literals $\neg A$ are all one.

Example 9. In this example we consider a BN2T with its structure defined in Figure 1, with CPTs corresponding to deterministic threshold with $k = 3$, $\ell = 2$, and with evidence $Y_1 = 1$ and $Y_2 = 1$, respectively. CPTs are decomposed as described in Example 8. We will show how this BN2T is encoded using a CNF. We will get following 68 logical variables, classified into two groups. The first group consists of 12 indicator variables:

$$\lambda^{x}_{X_i}, i \in \{1,2,3,4\}, x \in \{0,1\} \text{ and } \lambda^{y}_{Y'_j}, j \in \{1,2\}, y \in \{0,1\}.$$

The weights are all one for all 12 indicator variables and for both the positive and negative literals. The second group consists of 56 parameter variables:

$$\theta^{x}_{X_i}, i \in \{1,2,3,4\}, x \in \{0,1\}$$

$$\theta^{x,y}_{X_i,Y'_j}, (j,i) \in \{(1,1),(1,2),(1,3),(2,2),(2,3),(2,4)\}, x \in \{0,1\}, y \in \{1,2,3,4\}.$$

Each positive literal of a parameter variable has its weight defined to be the value of the corresponding parameter, i.e. for $i \in \{1,2,3,4\}, x \in \{0,1\}$ we have that $w(\theta^{x}_{X_i}) = P_{X_i}(x)$ and for $(j,i) \in \{(1,1),(1,2),(1,3),(2,2),(2,3),(2,4)\}$

$$w(\theta^{x,y}_{X_i,Y'_j}) = \sqrt[3]{\frac{1}{6}},\ \sqrt[3]{\frac{1}{3}},\ -\sqrt[3]{\frac{1}{2}},\ 2\sqrt[3]{\frac{1}{6}},\ -\sqrt[3]{\frac{1}{3}},\ 0$$

for $(x,y) = (0,1),(0,2),(0,3),(1,1),(1,2),(1,3)$, respectively. All negative literals have weight one. The CNF consits of 12 indicator clauses that ensure that in each evaluation exactly one of indicator variables of each variable is true, i.e., (15 holds for $i \in \{1,2,3,4\}$ and (16) holds for $j \in \{1,2\}$. Additionaly, there are 16 parameter clauses in the CNF, that ensure $\theta^x_{X_i}$ is true if and only if $\lambda^x_{X_i}$ is true, i.e., (17) holds for $i \in \{1,2,3,4\}$ and $x \in \{0,1\}$ and 144 parameter clauses, that ensure $\theta^{x,y}_{X_i,Y'_j}$ is true if and only if $\lambda^x_{X_i}$ and $\lambda^y_{Y'_j}$ is true, i.e., (18) holds for $(i,j) \in \{(1,1),(1,2),(1,3),(2,2),(2,3),(2,4)\}$, $x \in \{0,1\}$, and $y \in \{1,2,3,4\}$.

Theorem 1. *Let Δ be the logical theory of a BN2T with evidence e encoded by a CNF as specified above. Then for the weighted model count $w(\Delta)$ of the theory Δ it holds that*

$$w(\Delta) = P(e) \ . \tag{21}$$

Proof. In order to show that the WMC of the CNF encoded as specified above is equal to $P(e)$ we show how a weight of a logical model of the CNF of the BN2T is computed. The weight $w(\omega)$ of a logical model ω is the product of weights of its literals t:

$$w(\omega) = \prod_{\omega \models t} w(t) \ . \tag{22}$$

From the above encoding it follows that each model ω of the CNF corresponds to exactly one configuration of values of BN2T variables $X_i, Y'_j, i = 1,\ldots,n, j = 1,\ldots,m$. Without any loss of generality assume it is $(x_1,\ldots,x_n,y'_1,\ldots,y'_m)$. The weight of model ω is than equal to the product

$$w(t) = \left(\prod_{j=1}^{m} w(\lambda^{y'_j}_{Y'_j})\right) \left(\prod_{i=1}^{n} w(\lambda^{x_i}_{X_i}) w(\theta^{x_i}_{X_i})\right) \left(\prod_{j=1}^{m} \prod_{X_i \in pa(Y_j)}^{n} w(\theta^{x_i,y'_j}_{X_i,Y'_j})\right) \ . \tag{23}$$

The above CNF encoding assures that all other literals are negative and their weight is one. Therefore we could omit them from the formula (23). Substituting the values of weights into formula (23) we get

$$w(t) = \left(\prod_{i=1}^{n} P(x_i)\right) \left(\prod_{j=1}^{m} \prod_{X_i \in pa(Y_j)} \psi_{X_i,Y'_j}(x_i,y'_j)\right) \ . \tag{24}$$

The weight of a logical theory is the sum of weights of its logical models. If we sum the weights of all logical models of the CNF computed by formula (24) we can see that it is equal to formula (14), which is used to compute the probability of evidence $P(e)$. □

5. Summary and future work

In this paper we have used two-layered Bayesian networks with noisy threshold models to illustrate how the CP tensor decomposition can be combined with weighted model counting. In the similar manner the CP tensor decomposition can be utilized for other models of independence of causal influence and for general Bayesian networks.

In a near future we would like to perform experiments with BN2T using modified Cachet [19] in the same spirit as it was done by Wei Li et al. [17] for noisy-max. As it was noted in [11] it should be possible to convert Cachet into a compiler by modifying it to keep a trace of its operations. As a criteria for the comparisons we could use the number of operations performed by Cachet when computing the satisfying probability since this criteria characterizes well the computational complexity of inference with compiled arithmetic circuits. In this way it should be possible to make fair comparisons of not only different encodings as Chavira and Darwiche's encoding [18] and Sang, Beame, and Kautz's encoding [10] but also different WMC solvers as Cachet [19] and Ace [20].

References

[1] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[2] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs, 2nd ed.* Springer, 2007.

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

[4] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.

[5] David Heckerman and John S. Breese. A new look at causal independence. In R. Lopez de Mantaras and David Poole, editors, *Proc. of the Tenth Conf. on Uncertainty in AI*, pages 286–292, 1994.

[6] F. J. Díez and M. J. Druzdzel. Canonical probabilistic models for knowledge engineering. Technical Report CISIAD-06-01, UNED, Madrid, Spain, 2006.

[7] P. Savicky and J. Vomlel. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764, 2007.

[8] J. Vomlel and P. Tichavský. Computationally efficient probabilistic inference with noisy threshold models based on a CP tensor decomposition. In *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM 2012), Granada, Spain*, pages 355–362, September 19-21, 2012.

[9] F. J. Díez and S. F. Galán. An efficient factorization for the noisy MAX. *International Journal of Intelligent Systems*, 18:165–177, 2003.

[10] T. Sang, P. Beame, and H. Kautz. Solving Bayesian networks by weighted model counting. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05), Vol. 1*, pages 475–482. AAAI Press, 2005.

[11] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6):772–799, 2008.

[12] 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.

[13] Finn V. Jensen. *Bayesian networks and decision graphs*. Springer-Verlag, New York, 2001.

[14] S. Visscher, P. J. F. Lucas, C. A. M. Schurink, and M. J. M. Bonten. Modelling treatment effects in a clinical Bayesian network using Boolean threshold functions. *Artificial Intelligence in Medicine*, 46(3):251–266, 2009.

[15] M. A. J. van Gerven, R. Jurgelenaite, B. G. Taal, T. Heskes, and P. J. F. Lucas. Predicting carcinoid heart disease with the noisy-threshold classifier. *Artificial Intelligence in Medicine*, 40(1):45–55, 2007.

[16] R. Jurgelenaite, P. Lucas, and T. Heskes. Exploring the noisy threshold function in designing Bayesian networks. In M. Bramer, F. Coenen, and T. Allen, editors, *Research and Development in Intelligent Systems XXII*, pages 133–146. Springer, 2006.

[17] W. Li, P. Poupart, and P. van Beek. Exploiting structure in weighted model counting approaches to probabilistic inference. *Journal of Artificial Intelligence Research*, 40:729, 2011.

[18] A. Darwiche. A logical approach to factoring belief networks. In *Proceedings of International Conference on Knowledge Representation and Reasoning*, pages 409–420, 2002.

[19] Cachet. Model counting using component caching and clause learning. http://www.cs.rochester.edu/u/kautz/Cachet/index.htm, 2005.

[20] Ace. A Bayesian network compiler. http://reasoning.cs.ucla.edu/ace/, 2008.