# Approximating bounded tree-width Bayesian network classifiers with OBDD

**Karine Chubarian**[*] KCHUBA2@UIC.EDU
*University of Illinois Chicago*

**György Turán** [†] GYT@UIC.EDU
*University of Illinois Chicago, University of Szeged*

## Abstract

It is shown that Bayesian network classifiers of tree-width $k$ have an OBDD approximation computable in polynomial time in the parameters, for every fixed $k$. This is shown by approximating a polynomial threshold function representing the classifier. The approximation error can be measured with respect to any distribution which can be approximated by a mixture of bounded width distributions. This includes the input distribution of the classifier.

## 1. Introduction

A knowledge representation formalism is tractable if it allows for answering queries and for applying operations in polynomial time. The trade-off between expressivity (or succinctness) and tractability is an important consideration in many applications. The goal of *knowledge compilation* is to study the possibilities and limitations of compilation algorithms between different representation formalisms from the point of view of such trade-offs (Darwiche and Marquis (2002)).

In the Boolean case the *ordered binary decision diagram (OBDD)* representation is viewed as providing a good compromise between expressivity and tractability, and therefore it forms a standard representation or data structure for Boolean functions.

Chan and Darwiche (2003) introduced the problem of compiling Bayesian network classifiers into OBDD. They gave a compilation algorithm for the case of Naive Bayes Classifiers (NBC). For an NBC with $n$ input variables the size of the OBDD produced has size $O\left(2^{n/2}\right)$ and the running time of the algorithm is $O\left(n2^{n/2}\right)$. The compilation algorithm is extended to Latent-Tree Bayesian Network Classifiers in Shih et al. (2018b) producing OBDD of size $O\left(2^{3n/4}\right)$ in time $O\left(n2^{3n/4}\right)$. It is also shown that compiling even NBC to OBDD is NP-hard. A general compilation algorithm for Bayesian network classifiers is given in Shih et al. (2019), which has an exponential upper bound in the compilation width of the classifier. The compilation algorithms are used as a tool for reasoning about the classifiers and giving explanations for their decisions in Shih et al. (2018a), Darwiche and Hirth (2020) and Darwiche (2020). These applications provide a connection to interpretability.

OBDDs are related to the computational model of read-once branching programs, and the complexity of such models have been studied extensively in complexity theory. Hosaka et al. (1997) showed that linear threshold functions (LTF) can be computed by OBDD of size $O\left(2^{n/2}\right)$ and Takenaga et al. (1997) constructed an LTF such that every OBDD representing that function has

∗. Partially supported by the NSF grant CCF-1934915.
†. Partially supported by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008).

size $\Omega\left(2^{cn^{1-\varepsilon}}\right)$. Shih et al. (2018b) shows that if $P \neq NP$ then NBC cannot be compiled into OBDD in polynomial time (and thus having polynomial size), while Takenaga et al. (1997) give an exponential lower bound to the OBDD size needed, without any complexity-theoretic assumption.

The negative results raise the question whether there is a polynomial size *approximate* OBDD representation for NBC or perhaps even for larger classes of Bayesian network classifiers, and if so then can such an approximation be computed in polynomial time? The classifiers are usually assumed to be built using machine learning and so being only approximately correct. This gives a further reason to consider approximate compilations.

In this paper we answer this question positively by giving an approximation scheme for Bayesian network classifiers of bounded tree-width. In the context of knowledge compilation, this result corresponds to *approximate knowledge compilation*.

For approximate representations one has to specify what kind of approximation is considered. One option is to consider the number of truth assignments where the two representations differ, i.e., the error under the uniform distribution over truth assignments. However, a Bayesian network classifier already specifies a distribution on the input variables, namely the marginal distribution over these variables of the joint distribution represented by the classifier. It seems natural to use this distribution for measuring error. More generally, Theorem 1 applies to any distribution which can be approximated by a mixture of a bounded number of bounded-width distributions.

**Theorem 1** *For every Bayesian network classifier of tree-width $k$ having $n$ Boolean variables and $d$-bit conditional probabilities, and every $\varepsilon > 0$ there is an OBDD of size* $\mathsf{poly}\left(n^{O(k)}, d, 1/\varepsilon\right)$ *approximating the classifier with error at most $\varepsilon$ with respect to the input distribution of the classifier. The OBDD can be constructed in time* $\mathsf{poly}\left(n^{O(k)}, d, 1/\varepsilon\right)$.

This result generalizes a preliminary version of this paper (Chubarian and Turán (2020)) on Tree Augmented Bayesian Network Classifiers (TAN) to bounded tree-width classifiers. Negative results for approximate knowledge compilation were recently proved by de Colnet and Mengel (2020).

In the remainder of the introduction we give an outline of the proof. We use the fact that Bayesian network classifiers can be represented as *polynomial threshold functions (PTF)*. The terms of the polynomial, viewed as a hypergraph over the variables, have bounded path-width (Corollary 4). The compilation algorithm, then, is an approximate compilation of such PTF into OBDD.

For this compilation the underlying distribution over the input variables, measuring the error of the compilation, is assumed to be have a certain structural property called *(polynomially) bounded width*. Such syntactically defined discrete distributions were introduced by Kamp et al. (2006), and studied further in Gopalan et al. (2010). In these papers this class of distributions is called small-space sources (the definitions in the two papers are somewhat different). The same notion is considered in Bozga and Maler (1999) and Jaeger (2004).

The approximability result is the following.

**Theorem 2** *For every $n$-variable path-width-$k_1$ PTF $f$ with integer coefficients of at most $d$ bits, every width-$k_2$ distribution $D$ over the input variables with $d$-bit probabilities and every $\varepsilon > 0$ there is an OBDD of size* $\mathsf{poly}\left(n^{O(k_1)}, d, k_2, 1/\varepsilon\right)$ *approximating the PTF with error at most $\varepsilon$ with respect to $D$. The OBDD can be constructed in time* $\mathsf{poly}\left(n^{O(k_1)}, d, k_2, 1/\varepsilon\right)$.

The proof generalizes the OBDD construction of Gopalan et al. (2010) for approximating the number of solutions of knapsack problems, obtained by compressing the OBDD which works by evaluating the partial sums of the corresponding LTF.

The joint distribution of a Bayesian network classifier of bounded width is of bounded width itself, but this does not hold for the input distribution, which is the mixture of two bounded-width distributions. However, in Theorem 7 we show that the input distribution can be approximated by a bounded width distribution. Theorem 1 thus follows by applying Theorem 2 to this approximating distribution. The algorithms use width-based representation-finding and probabilistic inference algorithms having complexity $\mathsf{poly}\left(n, 2^k\right)$.

The paper is structured as follows. After discussing related work in the next section, Sections 3-4 give preliminaries. OBDDs for the distributions involved are described in Section 5, and for the OBDD computing the PTF exactly in Section 6. Section 7 contains the proofs of Theorems 2 and 1.

## 2. Related work

The representational power of probabilistic classifiers is discussed by Jaeger (2003) and Varando et al. (2015). General background for Bayesian networks, including compilation results are described in Darwiche (2009). Tractable operations and complexity aspects of OBDD are described in the monograph of Wegener (2000).

Complexity aspects of polynomials with bounded tree-width term structure are discussed in Makowsky and Meer (2000). Amarilli et al. (2018) give a path-width characterization of OBDD representability of monotone CNF. Levelwise monotonicity, a key OBDD property in Gopalan et al. (2010), is also used by Meka and Zuckerman (2010).

## 3. Preliminaries

### 3.1 Widths

For an undirected graph $G = (V, E)$, a *tree-decomposition* of $G$ is given by a tree $T$ and "bags" $B_t \subseteq V$ for every vertex $t$ of $T$, such that for every $(u, v) \in E$ there exists $t$ such that $u, v \in B_t$ and for every $v \in V$ the vertices $t$ such that $v \in B_t$ form a subtree of $T$. The *width* of a tree decomposition is $\max_{t \in V(T)} |B_t| - 1$ and the *tree-width* $\mathsf{tw}(G)$ of $G$ is the minimal width of tree-decompositions of $G$. The *path-width* $\mathsf{pw}(G)$ of $G$ is defined by trees restricted to paths.

The *moral graph* of a DAG $G$ is the undirected graph $\mathsf{MG}(G)$ obtained from $G$ by adding undirected edges between co-parents and disregarding the direction of the original directed edges. For a DAG $G$ we define $\mathsf{tw}(G) = \mathsf{tw}(\mathsf{MG}(G))$ and $\mathsf{pw}(G) = \mathsf{pw}(\mathsf{MG}(G))$. The undirected graph obtained from a DAG $G$ by disregarding the edge directions is denoted by $\mathsf{und}(G)$

The *primal graph* $\mathsf{PG}(H)$ of a hypergraph $H$ is the undirected graph obtained from $H$ by replacing every hyperedge by a clique. Then $\mathsf{tw}(H) = \mathsf{tw}(\mathsf{PG}(H))$ and $\mathsf{pw}(H) = \mathsf{pw}(\mathsf{PG}(H))$. Given an undirected graph $G$ with vertex set $[n] = \{1, \ldots, n\}$, the *separator* of vertex $\ell$ with respect to the natural ordering of the vertices is

$$\mathsf{S}_\ell = \{j \,:\, j \leq \ell \text{ and } (j, k) \in E \text{ for some } k > \ell\}. \tag{1}$$

We note that since the only new vertex which can enter $\mathsf{S}_\ell$ is $\ell$ we have

$$\mathsf{S}_\ell \subseteq \mathsf{S}_{\ell-1} \cup \{\ell\}. \tag{2}$$

The *vertex separation number* $\mathsf{vs}(G)$ of $G$ is the minimum of $\max_{\ell \leq n-1} |\mathsf{S}_\ell|$ over all orderings of the vertices. Kinnersley (1992) showed that $\mathsf{pw}(G) = \mathsf{vs}(G)$.

Let us recall several statements about path-width and tree-width.

**Claim 1** a) *For any $n > 2$ it holds that* $\mathsf{tw}(K_n) = n - 1$.

b) *For any (un)directed graph G and any $H \subseteq G$ it holds that* $\mathsf{pw}(H) \leq \mathsf{pw}(G)$, $\mathsf{tw}(H) \leq \mathsf{tw}(G)$.

c) *For any $n$-vertex (un)directed graph G it holds that* $\mathsf{pw}(G) = O(\mathsf{tw}(G) \log n)$.

d) *For every DAG G every vertex $v$ has at most* $\mathsf{tw}(G)$ *parents.*

### 3.2 Bayesian network classifiers

We use $X_i, x_i$, resp., $a_i$, to denote binary random variables, Boolean variables, resp., Boolean constants. A *Bayesian network classifier* $N$ is a DAG $G_N$ over binary *input variables* $X_1, \ldots, X_n$ and a binary *classifier variable* $C$, with local conditional probabilities specified for each vertex. It is assumed that $(C, X_i)$ is an edge for every $i = 1, \ldots, n$. Let $\Pi_i = \{j \,:\, X_j \text{ is a parent of } X_i\}$ be the set of parents of $X_i$ other than $C$ [1].

The *family* of $i$ is $\{i\} \cup \Pi_i$. Let $d_N = 1 + \max_i |\Pi_i|$ be the maximal size of families in $G_N$, referred to as the *degree* of the Bayesian network. The tree-width of $N$ is $\mathsf{tw}(G_N) = \mathsf{tw}(\mathsf{MG}(G_N \setminus \{C\}))$, its path-width is $\mathsf{pw}(G_N) = \mathsf{pw}(\mathsf{MG}(G_N \setminus \{C\}))$ and a separator $\mathsf{S}_\ell$ is computed using $\mathsf{MG}(G_N \setminus \{C\})$.

The restriction of a vector $x = (x_1, \ldots, x_n)$ to a subset $I$ of its coordinates is denoted by $x_I$. The local conditional probabilities are $p^0_c = P_N(C = c)$ and $p^i_{(a_i, a_{\Pi_i}, c)} = P_N(X_i = a_i \,|\, X_{\Pi_i} = a_{\Pi_i}, C = c)$ for $i = 1, \ldots, n$.

The joint distribution of the variables is

$$P_N(X_1 = a_1, \ldots, X_n = a_n, C = c) = p^0_c \prod_{i=1}^{n} p^i_{(a_i, a_{\Pi_i}, c)}. \tag{3}$$

The marginal distribution over the input variables, also referred to as the *input distribution*, is

$$P_{N,X}(X_1 = a_1, \ldots, X_n = a_n) = \sum_{c=0}^{1} P_N(X_1 = a_1, \ldots, X_n = a_n, C = c).$$

We also use simpler notations such as $P_N(a_1, \ldots, a_n, c)$ and $P_{N,X}(a_1, \ldots, a_n)$.

The Bayesian network classifier corresponding to $N$ is a Boolean function $f_N(x_1, \ldots, x_n)$ where $f_N(a_1, \ldots, a_n) = 1$ iff

$$P_N(a_1, \ldots, a_n, 1) \geq P_N(a_1, \ldots, a_n, 0).$$

### 3.3 OBDD and GOBDD

An *ordered binary decision diagram (OBDD)* over Boolean variables $x_1, \ldots, x_n$ computes a Boolean function $f$ [2]. An OBDD is a DAG with two sinks labeled 0 and 1, and the other nodes labeled with variables. The DAG is assumed to be layered, with directed edges going from a layer to the next layer, and sinks on the last layer. There are $n + 1$ layers and a permutation $\pi(i)$ of $[n]$ such that

1. Depending on the context, with an abuse of notation, we also consider the vertex sets to be $[n]$ or $\{x_1, \ldots, x_n\}$ throughout the paper.
2. The definition given here is that of a *complete* OBDD in Wegener (2000), a convenient slightly restricted version.

nodes on the $i$'th layer are labeled with variable $x_{\pi(i)}$. On the first layer there is a single start node labeled $x_{\pi(1)}$. Every non-sink node has two outgoing edges, labeled with 0, resp., 1. For every truth assignment $x = (x_1, \ldots, x_n)$, $f(x)$ is the label of the sink reached by following edge labels corresponding to the bits in $x$, evaluated in the order given by the labels of the layers. The *width* of an OBDD is the maximal number of nodes in a layer.

A *generator OBDD (GOBDD)* $D$ generates a probability distribution over $\{0,1\}^n$. It is similar to an OBDD, except edges are also labeled with probabilities, and there is a single sink. A probability $p_u$ is associated with every non-sink vertex $u$, and the 0-edge (resp. 1-edge) leaving $u$ is labeled $p_u^0 = p_u$ (resp., $p_u^1 = 1 - p_u$). For every truth assignment $x = (x_1, \ldots, x_n)$, the GOBDD determines a path from the source to the sink, and $P_D(x)$ is the product of edge probabilities along the path. The *width* of a GOBDD is the width if the underlying OBDD. The *width* of a distribution is the minimal width of GOBDDs generating it. Product distributions have width one.

## 4. Polynomial threshold function representation of Bayesian network classifiers

A *polynomial threshold function (PTF)* is a Boolean function of the form $\mathrm{sgn}(p(x_1, \ldots, x_n))$, where sgn is the sign function and $p$ is a multilinear polynomial. The degree of the representation is the degree of $p$, i.e., the maximal number of variables in a term. PTF can be written in general as

$$p(x_1, \ldots, x_n) = \sum_{I \in \mathcal{I}} \beta_I x_I, \tag{4}$$

where $\mathcal{I}$ is a family of subsets of $[n]$.

The *term-hypergraph* $H_p$ of the PTF $p$ has vertex set $[n]$ and edge set $\mathcal{I}$. The tree-width of a PTF is the tree-width of its term-hypergraph, and similarly for path-width and separator $\mathsf{S}_\ell$.

Varando et al. (2015) stated the representability of Bayesian network classifiers as PTF for categorical distributions, assuming non-zero conditional probabilities.

**Proposition 3** *(see, e.g., Varando et al. (2015)) Let $N$ be a Bayesian network classifier with non-zero conditional probabilities. The classifier $f_N$ is a degree-$d_N$ PTF such that every term is a subset of a family.*

**Proof** Let $I_a(x)$ be the binary indicator function, i.e., $I_1(x) = x$ and $I_0(x) = 1 - x$. It holds that

$$P_N(X_i = x_i \,|\, X_{\Pi_i} = x_{\Pi_i}, C = c) = \prod_{(a_i, a_{\Pi_i})} p^i_{(a_i, a_{\Pi_i}, c)}{}^{I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j)}.$$

Taking logarithms

$$\log P_N\left(X_i = x_i \,|\, X_{\Pi_i} = x_{\Pi_i}, C = c\right) = \sum_{(a_i, a_{\Pi_i})} \log\left(p^i_{(a_i, a_{\Pi_i}, c)}\right) \cdot I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j).$$

From (3) we get

$$\log P_N(X_1 = x_1, \ldots, X_n = x_n, C = c) = \log p^0_c + \sum_{i=1}^{n} \sum_{(a_i, a_{\Pi_i})} \log\left(p^i_{(a_i, a_{\Pi_i}, c)}\right) \cdot I_{a_i}(x_i) \prod_{j \in \Pi_i} I_{a_j}(x_j)$$

and the claim follows by the definition of $f_N$. ■

The proposition with Claim 1 implies the following.

**Corollary 4** *The classifier $f_N(x_1, \ldots, x_n)$ represented by a tree-width $k$ Bayesian network classifier with non-zero conditional probabilities is a PTF with path-width $O(k \log n)$.*

**Proof** For every edge $(i, j)$ of the primal graph of the term-hypergraph $H_p$ for the PTF $p$ constructed above there is a family of $N$ containing that edge, which then belongs to the moral graph of the classifier. ■

The primal graph may be a proper subset of the moral graph due to cancellations.

### 4.1 Zero handling and precision

Proposition 3 and Corollary 4 are valid in the general case allowing for zero conditional probabilities as well. Zero conditional probabilities can be replaced by sufficiently small numbers (chosen depending on the non-zero entries) without changing the classifier.

The classifier compares a sum of coefficients to zero. These coefficients are logarithms of conditional probabilities, and the probabilities are given with a certain precision (taking into consideration the elimination of zero conditional probabilities as well). One can show that it is enough to approximate logarithms of the probabilities with precision $\mathsf{poly}(n, d, k)$ to yield the same classifier. The approximations can be computed using Taylor series in time $\mathsf{poly}(n, d, k)$.

For the remainder of the paper it assumed that conditional probabilities are non-zero. We note that another approach, improving efficiency and interpretability, is to use the original product form without taking logarithms and handling information about zero probabilities symbolically.

## 5. Distribution widths and approximation

In this section we consider the widths of the joint, $C$-conditional and input distributions of a tree-width-$k$ Bayesian network classifier $N$. It is shown that for Bayesian network classifiers of bounded tree-width the joint and $C$-conditional distributions have bounded width and the input distribution can be approximated by a bounded width distribution.

Variables are evaluated according to a good path-width ordering. The following lemma gives the separator properties used in the computation (see also Theorem 4.1 in Darwiche (2009)). Recall that the separators $\mathsf{S}_\ell$ are computed with respect to the moral graph $\mathsf{MG}(G_N)$.

**Lemma 5** *For any assignment $\{a_1, \ldots, a_n, c\} \in \{0, 1\}^{n+1}$ and any $\ell$ it holds that*

$$P_N\left(a_\ell \,|\, a_1, \ldots, a_{\ell-1}, c\right) = P_N\left(a_\ell \,|\, a_{\mathsf{S}_{\ell-1}}, c\right).$$

**Proof** We need to show that for every $m \in [\ell-1] \setminus \mathsf{S}_{\ell-1}$ the set $\{C, X_{\mathsf{S}_{\ell-1}}\}$ is a $d$-separator between $X_m$ and $X_\ell$ in $G_N$. Consider a path $P$ in $\mathsf{und}(G_N)$ between $X_m$ and $X_\ell$. If $P$ contains $C$ then it is blocked by $C$ as valves $X_i - C - X_j$ are divergent.

Assume that $P$ does not contain $C$. Let $X_t$ be the first vertex on $P$ with $t \geq \ell$. Let $Q$ be the initial segment of $P$ ending at $X_t$. Every inner node $X_i$ in $Q$ has $i < \ell$. It holds that $|Q| \geq 2$ as

otherwise $m \in \mathsf{S}_{\ell-1}$. Let the last two edges of $Q$ be $(X_{s_2}, X_{s_1})$ and $(X_{s_1}, X_t)$, and consider the valve $\mathbf{v} = X_{s_2} - X_{s_1} - X_t$. Here $s_1 \in \mathsf{S}_{\ell-1}$ by definition.

There are three cases. If $\mathbf{v}$ is sequential or divergent then it is closed due to $X_{s_1}$ and therefore $P$ is blocked. If $\mathbf{v}$ is convergent then $X_{s_2} \in \Pi(X_{s_1})$. But then $s_2$ and $t$ are co-parents and therefore $(s_2, t)$ is an edge of $\mathsf{MG}(G_N)$, and so $s_2 \in \mathsf{S}_{\ell-1}$ as well. Then $X_{s_2}$ is also an inner node of $Q$. Its valve $\mathbf{v}' = X_{s_3} - X_{s_2} - X_{s_1}$ is either sequential or divergent due to $(X_{s_2}, X_{s_1}) \in E(G_N)$, and hence it is closed due to $X_{s_2}$. Thus $P$ is blocked in this case as well. ■

Lemma 5 implies that the joint distribution can be written as

$$P_N(X_1 = a_1, \ldots, X_n = a_n, C = c) = p_c^0 \prod_{\ell=2}^{n} P_N\left(a_\ell \,|\, a_{\mathsf{S}_{\ell-1}}, c\right). \tag{5}$$

**Lemma 6** *Let $N$ be a tree-width-$k$ Bayesian network classifier. Then the joint distribution $P_N(X_1, \ldots, X_n, C)$ and the conditional distributions $P_N(X_1, \ldots, X_n|C)$ have width $n^{O(k)}$.*

**Proof** Let $N$ be a tree-width-$k$ Bayesian network classifier. Then by Claim 1 and the result of Kinnersley (1992) there is an ordering of the input variables such that every extended separator has size $O(k \log n)$. We use this ordering to construct a GOBDD computing the joint distribution.

The GOBDD $J$ has $n+2$ levels; the zero level corresponds to $C$ and has a single node $u_\emptyset^0$. It has two outgoing edges labelled with $p_0, p_1$, each edge leads to a child $u_0^1$ and $u_1^1$, respectively. Level $\ell$ evaluates $x_\ell$ and contains $2^{|\mathsf{S}_{\ell-1}|+1}$ nodes denoted as $u_{c,b}^\ell$, corresponding to truth assignments to $c$ and $x_{\mathsf{S}_{\ell-1}}$. (As the GOBDD has the classifier variable coming first, we now switch from the previous notation of put $c$ first.) We set $\mathsf{S}_0 = \emptyset$. Each $u_{c,b}^\ell$ has two outgoing edges evaluating $x_\ell = j$ with $j \in \{0, 1\}$; the edges are labelled with $p_{j,c,b}^\ell = P_N(x_\ell = j \,|\, x_{\mathsf{S}_{\ell-1}} = b, C = c)$. For $\ell < n$ the children of $u_{c,b}^\ell$ are $u_{c,b_0}^{\ell+1}$ and $u_{c,b_1}^{\ell+1}$ which correspond to assigning $x_\ell = 0$ or $x_\ell = 1$ respectively. The assignments $b_j$ with $j \in \{0, 1\}$ correspond to the assignment of $\mathsf{S}_\ell$. By (2), $b_j$ can be updated using $b$ and $j$. At level $n$, all edges go to the sink. The correctness of $J$ follows from (5).

The sub-GOBDD's starting at the children of the root represent the conditional distributions. The size bound for the GOBDD's follows from Claim 1. ■

The input distribution $P_{N,X}(a_1, \ldots, a_n)$ can be written as

$$P_N(a_1, \ldots, a_n \,|\, 0)\, P_N(0) + P_N(a_1, \ldots, a_n \,|\, 1)\, P_N(1),$$

so by Lemma 6 it is a mixture of two polynomial-width distributions. It is not necessarily of polynomial width itself (see Shen et al. (2016)). On the other hand, we now show that it can be approximated by a polynomial-width distribution.

The edge probabilities in a GOBDD for the input distribution can be written as

$$P_N(a_\ell \,|\, a_1, \ldots, a_{\ell-1}) = \frac{P_N(a_1, \ldots, a_\ell)}{P_N(a_1, \ldots, a_{\ell-1})} = \frac{\sum_{c=0}^{1} P_N(a_1, \ldots, a_\ell, c)}{\sum_{c=0}^{1} P_N(a_1, \ldots, a_{\ell-1}, c)}. \tag{6}$$

The partial truth assignments in the last expression correspond to paths in the GOBDD $J$ of Lemma 6, beginning with the start node. This suggests approximating $P_N(a_\ell \,|\, a_1, \ldots, a_{\ell-1})$ by approximating the terms on the right. This can be achieved by extending $J$ with the required approximations.

**Theorem 7** *There is a distribution $D(X_1, \ldots, X_n)$ of width* poly $\left(n^{O(k)}, d, 1/\varepsilon\right)$ *such that for every $a = (a_1, \ldots, a_n)$ it holds that*

$$(1-\varepsilon)P_D(a) \leq P_{N,X}(a) \leq (1+\varepsilon)P_D(a).$$

**Proof** First we describe an auxiliary construction $J'$, which adds approximate evaluation of probabilities assigned to paths beginning at the start node of $J$, as suggested by (6).

### 5.1 Description of $J'$

Levels are $0, \ldots, n+1$. Nodes are of the form $(u, r)$, where $u$ is a node of $J$ (including the sink), and $r \in \mathbb{N}$ (possible values for $r$ are bounded by a polynomial in the parameters).

Assume that a partial truth assignment $C = c, X_1 = a_1, \ldots, X_{\ell-1} = a_{\ell-1}$ ends at $u$ on level $\ell$ in $J$ and thus the product of edge probabilities along its path is $P_N(c, a_1, \ldots, a_{\ell-1})$. Then in $J'$ it ends in $(u, r)$ where

$$(1-\delta)^{r+\ell} < P_N(c, a_1, \ldots, a_{\ell-1}) \leq (1-\delta)^r \tag{7}$$

for a parameter $\delta$ to be determined later. Thus each node in $J$ is split into several copies, collecting paths with similar probabilities. If $u = u^{\ell}_{c,b}$ is a node in $J$ on level $\ell$ then the children of $(u, r)$ in $J'$ are

$$\left(u^j, r + \lfloor \log_{1-\delta} p^j_u \rfloor\right), \tag{8}$$

where $u^j$ is the $j$-child of $u$ in $J$, and $p^j_u = p^{\ell}_{j,c,b}$ is the probability of the edge $(u, u^j)$. Then (7) follows by induction, considering $(1-\delta)^{t+1} < p^j_u \leq (1-\delta)^t$ for some $t$.

The OBDD $D$ approximating $P_{N,X}$ is built using $J'$. Note that the polynomial dependence on $d$ appears through the inclusion of probabilities in (8).

### 5.2 Description of $D$

The levels are now $1, \ldots, n+1$, with $x_1$ evaluated on level 1. Nodes on level $\ell$ are $(v_0, v_1)$, where $v_0, v_1$ are level $\ell$ nodes of $J'$. The start node is $(v^*_0, v^*_1)$, where $v^*_0, v^*_1$ are the children of the start node in $J'$. The $j$-child of $(v_0, v_1)$ in $D$ is $\left(v^j_0, v^j_1\right)$, where $v^j_i$ is the $j$-child of $v_i$ in $J'$. The sinks of $J'$ are combined into a single sink.

If $v_0 = (u_0, r_0)$ and $v_1 = (u_1, r_1)$ with children $v^j_i = \left(u^j_i, r^j_i\right)$ then the probability of the edge from $(v_0, v_1)$ to its $j$-child $(v^j_0, v^j_1)$ is defined as the approximation of (6), i.e.,

$$\frac{(1-\delta)^{r^j_0} + (1-\delta)^{r^j_1}}{(1-\delta)^{r_0} + (1-\delta)^{r_1}}.$$

For the approximation property of the distribution generated by $D$, consider a partial truth assignment $a_1, \ldots, a_\ell$. Let the nodes reached by the extensions $(c, a_1, \ldots, a_{\ell-1})$ in $J'$ be $(u_0, r_0), (u_1, r_1)$, and those reached by $(c, a_1, \ldots, a_\ell)$ be $\left(u^{a_\ell}_0, r^{a_\ell}_0\right), \left(u^{a_\ell}_1, r^{a_\ell}_1\right)$. Then (6) and (7) imply that

$$\frac{(1-\delta)^{r^{a_\ell}_0+(\ell+1)} + (1-\delta)^{r^{a_\ell}_1+(\ell+1)}}{(1-\delta)^{r_0} + (1-\delta)^{r_1}} \leq P_N(a_\ell \,|\, a_1, \ldots, a_{\ell-1}) \leq \frac{(1-\delta)^{r^{a_\ell}_0} + (1-\delta)^{r^{a_\ell}_1}}{(1-\delta)^{r_0+\ell} + (1-\delta)^{r_1+\ell}} \tag{9}$$

Combining this with

$$\frac{(1-\delta)^{r_0^{a_\ell}} + (1-\delta)^{r_1^{a_\ell}}}{(1-\delta)^{r_0} + (1-\delta)^{r_1}} = P_D\left(a_\ell \,|\, a_1, \ldots, a_{\ell-1}\right),$$

and multiplying the inequalities it follows that

$$(1-\delta)^{n(n+1)} P_D(a) \leq P_{N,X}(a) \leq P_D(a)(1-\delta)^{-n(n+1)}$$

for every truth assignment $a$. Thus the theorem follows with choosing $\delta = \Theta\left(\varepsilon/n^2\right)$. ■

The same proof shows that, in general, the mixture of a constant number of bounded-width distributions can be approximated by a bounded-width distribution with a polynomial width blow-up.

## 6. Exact OBDD for a PTF

In this section we describe an OBDD $B$ computing a PTF exactly. $B$ also computes acceptance probabilities at each node *w.r.t.* a distribution $D$ over $\{0,1\}^n$ represented by a GOBDD. The size of $B$ is exponential. The probabilities will be used in the next section, where $B$ is compressed, introducing a small error with respect to $D$. The compression process uses $B$ in a "virtual" manner.

Let $f(x_1, \ldots, x_n) = \operatorname{sgn}(p(x_1, \ldots, x_n))$ be a PTF $p$ of the form (4) with integer coefficients and let $D$ be a distribution represented by a GOBDD over $\{0,1\}^n$. Let $W$ be an upper bound for the sum of the absolute values of the coefficients of $p$. There are $n+1$ layers; for $2 \leq \ell \leq n$ the layer $\ell$ contains nodes labelled as $v^\ell_{s,b,u}$ where $-W \leq s \leq W$, $b \in \{0,1\}^{|\mathsf{S}_{\ell-1}|}$ is an assignment to the bits in $x_{\mathsf{S}_{\ell-1}}$ and $u$ is a node of $D$ on layer $\ell$. For $\ell = 1$ there is one node $v^1_{0,\emptyset,u^1}$ where $u^1$ is the start node of $D$.

The children of a node $v^\ell_{s,b,u}$ are $v^{\ell+1}_{s_0,b_0,u_0}$ and $v^{\ell+1}_{s_1,b_1,u_1}$; they correspond to the evaluations $x_\ell = 0$ and $x_\ell = 1$. The nodes $u_0$ and $u_1$ are 0 and 1 child of a node $u$ in $D$. The values $s_0, s_1$ are determined as follows. First, let

$$\mathsf{sum}(b, x_\ell) = \left( \sum_{I \in \mathcal{I}: I \subseteq \mathsf{S}_{\ell-1} \cup \{\ell\}, \ell \in I} \beta_I b_{I \setminus \{\ell\}} \right) x_\ell$$

This sum represents the terms which become constant when $x_\ell$ is evaluated, assuming that their other variables are assigned truth values according to $b$. Note that the use of the primal graph guarantees that all such terms are accounted for in $\mathsf{S}_\ell$. Then we set $s_0 = s + \mathsf{sum}(b, 0), s_1 = s + \mathsf{sum}(b, 1)$. The updates to $b_0$ and $b_1$ are done using (2).

On the last level there are nodes of the form $v^{n+1}_{s,\emptyset,u^{n+1}}$ where $u^{n+1}$ is a sink of $D$. Nodes with $s < 0$ (resp., $s \geq 0$) are replaced by the sink nodes $\mathsf{sink}_0, \mathsf{sink}_1$ which reject or accept respectively.

For a node $v^\ell_{s,b,u}$ with $1 \leq \ell \leq n$ its *acceptance set* $A^\ell_{s,b,u} \subseteq \{0,1\}^{n-\ell+1}$ is the set of all assignments to $x_\ell, \ldots, x_n$ which, when starting at that node lead to $\mathsf{sink}_1$ in $B$. Acceptance sets are computed recursively. We set $A^{n+1}_{\mathsf{sink}_0} = \emptyset$ and $A^{n+1}_{\mathsf{sink}_1} = \{\epsilon\}$ where $\epsilon$ is the empty string. For $1 \leq \ell \leq n$ it holds that $A^\ell_{s,b,u} = 0 \cdot A^{\ell+1}_{s_0,b_0,u_0} \cup 1 \cdot A^{\ell+1}_{s_1,b_1,u_1}$, where $\cdot$ denotes string concatenation.

Furthermore, given a node $v^\ell_{s,b,u}$ for $1 \leq \ell \leq n$ its *acceptance probability* $P_D\left(v^\ell_{s,b,u}\right)$ is the probability that a random truth assignment to $x_\ell, \ldots, x_n$ leads from $v^\ell_{s,b,u}$ to $\mathsf{sink}_1$ in $B$. The

probabilities are evaluated in $D$ starting from $u$. We set $P_D(\mathsf{sink}_0) = 0$ and $P_D(\mathsf{sink}_1) = 1$. For $\ell \leq n$ we have

$$P_D\left(v^\ell_{s,b,u}\right) = p^0_u P_D\left(v^{\ell+1}_{s_0,b_0,u_0}\right) + p^1_u P_D\left(v^{\ell+1}_{s_1,b_1,u_1}\right).$$

**Lemma 8** *Let $v^\ell_{s_1,b,u}, v^\ell_{s_2,b,u} \in B$ be such that $s_1 < s_2$. Then*

a) $A^\ell_{s_1,b,u} \subseteq A^\ell_{s_2,b,u}$,

b) $P_D\left(v^\ell_{s_1,b,u}\right) \leq P_D\left(v^\ell_{s_2,b,u}\right)$.

## 7. Proof of Theorems 2 and 1

First we prove Theorem 2. The approximate OBDD $\widetilde{B}$ satisfying the requirements is constructed by compressing $B$, processing its levels from the bottom up and within each level from left to right. We set $W = n^{k+1}\, 2^{d+2}$. Each level of $B$ is partitioned into *blocks*

$$\mathsf{BL}^\ell_{b,u} = \{v^\ell_{s,b,u} : -W \leq s \leq W\}.$$

In each block a polynomial size set of *distinguished nodes* $\mathsf{DN}^\ell_{b,u} \subseteq \mathsf{BL}^\ell_{b,u}$ is selected. These are the nodes of $\widetilde{B}$. The children of distinguished nodes are modified to be the closest distinguished node with a larger $s$-value. The processing of a level also includes the calculation of modified acceptance probabilities $\tilde{P}_D\left(v^\ell_{s,b,u}\right)$, using the modified acceptance probabilities of the modified children. $B$ is used implicitly, by doing binary search on the $s$ values.

The construction of $\widetilde{B}$ uses the procedure $\mathsf{BUILD}\left(v^\ell_{s,b,u}\right)$.

**Procedure** $\mathsf{BUILD}\left(v^\ell_{s,b,u}\right)$

**if** $\ell = n$ **then** children and acceptance probabilities are unchanged

**else**

modify the children: the new 0-child is $v^{\ell+1}_{s',b_0,u_0}$, resp. the new 1-child is $v^{\ell+1}_{s'',b_1,u_1}$, where

$$s' = \min\left\{t \,:\, v^{\ell+1}_{t,b_0,u_0} \in \mathsf{DN}^{\ell+1}_{b_0,u_0}, s_0 \leq t\right\}, \quad s'' = \min\left\{t \,:\, v^{\ell+1}_{t,b_1,u_1} \in \mathsf{DN}^{\ell+1}_{b_1,u_1}, s_1 \leq t\right\},$$

compute the new acceptance probability $\tilde{P}_D\left(v^\ell_{s,b,u}\right) = p^0_u \tilde{P}_D\left(v^{\ell+1}_{s',b_0,u_0}\right) + p^1_u \tilde{P}_D\left(v^{\ell+1}_{s'',b_1,u_1}\right)$.

Applying the procedure $BUILD$ repeatedly, we find the set of distinguished vertices $\mathsf{DN}^\ell_{b,u}$ with $s$-values $s^*_0 < s^*_1 < \ldots$, where $s^*_0 = -W$ and

$$s^*_{i+1} = \begin{cases} s^*_i + 1 \text{ if } \tilde{P}_D\left(v^\ell_{s^*_i+1,b,u}\right) > (1+\delta)\tilde{P}_D\left(v^\ell_{s^*_i,b,u}\right) \\ \max\left\{t \,:\, \tilde{P}_D\left(v^\ell_{t,b,u}\right) \leq (1+\delta)\tilde{P}_D\left(v^\ell_{s^*_i,b,u}\right)\right\} \quad \text{else,} \end{cases} \tag{10}$$

where $\delta$ is to be specified later. After all the distinguished sets are constructed, there may be nodes not reachable from the start node; they are removed by one pass from the start node.

Note that it follows by induction from the definition of children that $\tilde{P}(v^\ell_{s,b,u})$ is monotonic in $s$. Let $\tilde{A}^\ell_{s,b,u}$ denote the acceptance sets in $\widetilde{B}$.

**Lemma 9** *For any level $\ell \leq n$ and any distinguished node $v^{\ell}_{s,b,u} \in \mathsf{DN}^{\ell}_{b,u}$ it holds that*

*a)* $A^{\ell}_{s,b,u} \subseteq \tilde{A}^{\ell}_{s,b,u}$,

*b)* $P_D\left(v^{\ell}_{s,b,u}\right) \leq \tilde{P}_D\left(v^{\ell}_{s,b,u}\right) \leq (1+\delta)^{n-\ell} P_D\left(v^{\ell}_{s,b,u}\right)$.

**Proof** Part *a)* follows from Lemma 8 by noting that the $s$-values are always increased. For part *b)*, the first inequality follows from *a)*. For the second inequality we claim that

$$\tilde{P}_D\left(v^{\ell+1}_{s',b_0,u_0}\right) \leq (1+\delta)\tilde{P}_D\left(v^{\ell+1}_{s_0,b_0,u_0}\right), \tag{11}$$

where $s_0$ is the 0-child of $s$ in $B$, and $s'$ is its new 0-child found by procedure BUILD. This holds by definition if $v^{\ell+1}_{s',b_0,u_0}$ is included in $\mathsf{DN}^{\ell+1}_{b_0,u_0}$ using the second case in (10). Otherwise $s_0 = s'$, so the claim holds again. The analogous statement holds for 1 instead of 0 as well.

Using the definition of $\tilde{P}_D$, (11) and induction

$$\begin{aligned}\tilde{P}_D\left(v^{\ell}_{s,b,u}\right) &= p^0_u \tilde{P}_D\left(v^{\ell+1}_{s',b_0,u_0}\right) + p^1_u \tilde{P}_D\left(v^{\ell+1}_{s'',b_1,u_1}\right) \leq (1+\delta)\left(p^0_u \tilde{P}_D\left(v^{\ell+1}_{s_0,b_0,u_0}\right) + p^1_u \tilde{P}_D\left(v^{\ell+1}_{s_1,b_1,u_1}\right)\right)\\ &\leq (1+\delta)^{n-\ell}\left(p^0_u P_D\left(v^{\ell+1}_{s_0,b_0,u_0}\right) + p^1_u P_D\left(v^{\ell+1}_{s_1,b_1,u_1}\right)\right) = (1+\delta)^{n-\ell} P_D\left(v^{\ell}_{s,b,u}\right).\end{aligned}$$

■

By Lemma 9 *b)* choosing the value $\delta = O(\varepsilon/n)$ gives that for the root $v^1_{0,\emptyset,\emptyset}$ it holds that $P\left(v^1_{0,\emptyset,\emptyset}\right) \leq \tilde{P}\left(v^1_{0,\emptyset,\emptyset}\right) \leq (1+\varepsilon)P\left(v^1_{0,\emptyset,\emptyset}\right)$. Lemma 9 *a)* implies that $\widetilde{B}$ has one-sided error at most $\varepsilon$ w.r.t. $D$.

The monotonicity of acceptance probabilities implies that the next distinguished node $s^*_{i+1}$ can be found by binary search over $[s^*_i, W]$, calling BUILD in each step to compute $\tilde{P}$. Binary search is polynomial in the parameters. The number of distinguished nodes is also polynomial, as their $s$-values increase exponentially. Thus the size of the OBDD and the running time are both polynomial.

Theorem 1 follows from applying Theorem 2 to the PTF of Bayesian network classifier $N$ and distribution $D$ approximating input distribution $P_{N,X}$, using error parameter $\varepsilon/3$ in both cases.

## References


A. Amarilli, M. Monet, and P. Senellart. Connecting Width and Structure in Knowledge Compilation. In *ICDT, 2018*, volume 98 of *Leibniz Int. Proc. in Inf. (LIPIcs)*, pages 6:1–6:17, 2018.

M. Bozga and O. Maler. On the representation of probabilities over structured domains. In *11th Computer Aided Verification, CAV '99*, volume 1633 of *LNCS*, pages 261–273. Springer, 1999.

H. Chan and A. Darwiche. Reasoning about Bayesian network classifiers. In *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 107–115, 2003.

K. Chubarian and G. Turán. Interpretability of Bayesian network classifiers: OBDD approximation and polynomial threshold functions. In *ISAIM*, 2020.

A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Camb. Univ. Press, 2009.

A. Darwiche. Three modern roles for logic in AI. *CoRR*, abs/2004.08599, 2020.

A. Darwiche and A. Hirth. On the reasons behind decisions. *CoRR*, abs/2002.09284, 2020.

A. Darwiche and P. Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264, 2002.

A. de Colnet and S. Mengel. Lower bounds for approximate knowledge compilation. In C. Bessiere, editor, *Proc. of the Twenty-Ninth Int. Joint Conf. on AI 2020*, pages 1834–1840, 2020.

P. Gopalan, A. R. Klivans, and R. Meka. Polynomial-time approximation schemes for knapsack and related counting problems using branching programs. *CoRR*, abs/1008.3187, 2010.

K. Hosaka, Y. Takenaga, T. Kaneda, and S. Yajima. Size of ordered binary decision diagrams representing threshold functions. *Theor. Comput. Sci.*, 180(1-2):47–60, 1997.

M. Jaeger. Probabilistic classifiers and the concepts they recognize. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 266–273, 2003.

M. Jaeger. Probabilistic decision graphs - combining verification and AI techniques for probabilistic inference. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 12(Supplement-1):19–42, 2004.

J. Kamp, A. Rao, S. P. Vadhan, and D. Zuckerman. Deterministic extractors for small-space sources. In *STOC, 2006*, pages 691–700, 2006.

N. G. Kinnersley. The vertex separation number of a graph equals its path-width. *Inf. Process. Lett.*, 42(6):345–350, 1992.

J. Makowsky and K. Meer. Polynomials of bounded tree-width. In *Formal Power Series and Algebraic Combinatorics*, pages 692–703. Springer, 2000.

R. Meka and D. Zuckerman. Pseudorandom generators for polynomial threshold functions. In *STOC 2010*, pages 427–436, 2010.

Y. Shen, A. Choi, and A. Darwiche. Tractable operations for arithmetic circuits of probabilistic models. In *NIPS, 2016*, pages 3936–3944, 2016.

A. Shih, A. Choi, and A. Darwiche. Formal verification of Bayesian network classifiers. In M. Studený and V. Kratochvíl, editors, *PGM 2018*, volume 72 of *Proceedings of Machine Learning Research*, pages 427–438. PMLR, 2018a.

A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining Bayesian network classifiers. In *IJCAI 2018*, pages 5103–5111, 2018b.

A. Shih, A. Choi, and A. Darwiche. Compiling Bayesian network classifiers into decision graphs. In *AAAI 2019,*, pages 7966–7974, 2019.

Y. Takenaga, M. Nouzoe, and S. Yajima. Size and variable ordering of OBDDs representing threshold functions. In *Computing and Combinatorics*, pages 91–100, 1997.

G. Varando, C. Bielza, and P. Larrañaga. Decision boundary for discrete Bayesian network classifiers. *Journal of Machine Learning Research*, 16:2725–2749, 2015.

I. Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM, 2000.