Learning NAT-Modeled Bayesian Network Structures with Bayesian Approach

Yang Xiang*, Wanrong Sun
University of Guelph, Canada

Abstract

We study Bayesian approach for learning structures of Bayesian networks (BNs) with local models. The local structures we focus on are Non-impeding noisy-AND Tree (NAT) models due to their multiple merits. We extend meta-nets to allow encoding of prior knowledge on NAT local structures and parameters. From the extended meta-nets, we develop a Bayesian Dirichlet (BD) scoring function for evaluating alternative NAT-modeled BN structures. A heuristic algorithm is presented for searching through the structure space that is significantly more complex than that of BN structures without local models. We experimentally demonstrate learning of NAT-modeled BNs, whose inference produces sufficiently accurate posterior marginals and is significantly more efficient.

Keywords: Uncertainty, Machine Learning, Learning Bayesian Nets, Non-impeding Noisy-AND Trees, Local Models

1. Introduction

Learning BNs from data is an important task in probabilistic reasoning. BNs avoid combinatorial explosion on the number of variables by encoding conditional independence in graphical structures, but space and inference time grow exponentially in the number of causes per effect due to tabular conditional probability distributions (CPDs). To overcome this limitation of *tabular BNs*, local models have been applied, such as noisy-OR, noisy-MAX [1], context-specific independence (CSI) [2], DeMorgan [3], tensor-decomposition [4], and cancellation [5]. Merits of local models lead to learning BNs with local structures.

We focus on NAT local models [6] due to several merits: simple causal interactions (reinforcement/undermining), expressiveness (recursive mixture of causal interactions, multi-valued, ordinal or nominal [7]), generality (generalizing noisy-OR, noisy-MAX, and DeMorgan), and orthogonality to CSI. While tabular BN inference is exponential in treewidth, inference is tractable with NAT-modeled BNs of high treewidth and low density. In particular, space of a tabular BN (measured by the total number of CPD parameters) is $O(Ks^n)$, where K is the number of variables, s bounds domain sizes of variables, and n bounds numbers of causes (parents) per variable. In fully NAT-modeled BNs (see Section 2.2), dependencies of variables on their parents are quantified by NAT models instead of tabular CPDs, resulting in $O(K\ s\ n)$ space. This efficiency extends to inference when NAT-modeled BNs have structures of high treewidth (lower-bounded by n) and low density (measured by percentage of arcs beyond being singly connected) [1].

A large literature exists on learning tabular BNs, e.g., [8–12]. A common method is to combine heuristic search with a scoring function, where MDL [9] and BD [8, 10, 11] scores are often applied. This work focus on learning NAT-modeled BNs, due to their above merits. A recent work [13] enables learning NAT-modeled BNs based on MDL scores. The contribution of this work is a novel framework for learning structures of NAT-modeled BNs from data based on (extended) BD scores.

[1] Low density itself does not imply tractability: Tree tabular BNs (low density) of large n (large treewidth) are exponential in n.

* yxiang@uoguelph.ca

In the remainder, Section 2 reviews backgrounds on BD scores for learning tabular BN structures and on NAT-modeled BNs. Section 3 introduces the task of learning NAT-model BN structures with BD scores. Sections 4 through 6 present component BD subscores on likelihood, local structure prior, and global structure prior. Section 7 describes the heuristic search algorithm and analyzes its complexity. Experimental study is reported in Section 8.

2. Background

2.1. BD Scores for Learning Tabular BN Structures

A tabular BN over a set V of variables has a structure G and a collection Θ of parameter sets. G is a directed acyclic graph (DAG), whose nodes are labeled by variables in V. Each $x \in V$ and its parents π in G forms a *family*. Dependency of x on π is specified by a set of CPDs, with one CPD $Pr(x|\pi = \tau)$ per instantiation τ of π. Each parameter set $\theta_{x|\tau} \in \Theta$ specifies a CPD $Pr(x|\pi = \tau)$ (as *domain knowledge*).

The Bayesian approach to structure learning integrates *prior knowledge* (denote by $P()$) on G and Θ with data D: $P()$ expresses subjective probabilistic knowledge about the probabilistic domain knowledge expressed through $Pr()$. We assume that D has $N = |D|$ records on V, is *complete* (no missing value), and is *exchangeable* [10]. Given a candidate structure G, the probability $P(G, D) = P(G)P(D|G)$ is evaluated, where $P(G)$ (*structure prior*) encodes prior knowledge on G. Likelihood $P(D|G)$ can be evaluated using a *meta-net* Φ, derived from the *base-net* G and data D, which integrates prior knowledge on Θ with data D. For each $\theta_{x|\tau} \in \Theta$, there is a variable in Φ, which we denote also by $\theta_{x|\tau}$. Prior knowledge on $Pr(x|\pi = \tau)$ is encoded by a probability density function (pdf) $\rho(\theta_{x|\tau})$. For each record $d^i \in D$ (superscripts index records), meta-net Φ contains an instance G^i of base-net G. For each $x \in V$, besides π from G, x in G^i has extra parents $\theta_{x|\tau}$, one per instantiation of π.

Fig. 1 (a) shows a base-net, where $V = \{a, b\}$, $a \in \{a_0, a_1\}$, and $b \in \{b_0, b_1\}$. Without confusion, when variables are differentiated by symbols, subscripts index variable values, and when variables are differentiated by 1st subscripts, 2nd subscripts index values. Since $|D| = 2$, the meta-net in (b) contains G^1 (including a^1 and b^1) and G^2. Meta-net topology (no direct arcs among θ nodes) encodes the *parameter independence* assumption [11].

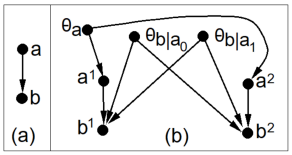

Figure 1. (a) Base-net. (b) Meta-net.

We assume that prior pdf $\rho(\theta_{x|\tau})$ is Dirichlet. In particular, let s be the domain size of x, i.e., $x \in \{x_1, ..., x_s\}$. Hence, $\theta_{x|\tau} = \{\theta_{x_1|\tau}, ..., \theta_{x_s|\tau}\}$ with $\sum_{i=1}^{s} \theta_{x_i} = 1$. The Dirichlet pdf with integer exponent parameters $\psi_{x_i|\tau}$ $(i = 1, ..., s)$ can be written as

$$\rho(\theta_{x|\tau}) = \eta \prod_i (\theta_{x_i|\tau})^{(\psi_{x_i|\tau})-1},$$

where η is a normalizing constant. Sum $\psi_{x|\tau} = \sum_{i=1}^{s} \psi_{x_i|\tau}$ is the *equivalent sample size*.

Given a base-net G, Dirichlet priors on parameters in Θ, and complete data D, using the meta-net with D as evidence, likelihood $P(D|G)$ can be evaluated [8, 11] as

$$P(D|G) = \prod_{x \in V} \prod_{\tau} \frac{\Gamma(\psi_{x|\tau})}{\Gamma(\psi_{x|\tau} + \sharp(\tau))} \prod_{\chi} \frac{\Gamma(\psi_{\chi|\tau} + \sharp(\chi, \tau))}{\Gamma(\psi_{\chi|\tau})}, \tag{2.1}$$

where $\Gamma()$ is the Gamma function, $\sharp(\chi, \tau)$ counts records of D that instantiate family of x to $(x = \chi, \pi = \tau)$, and $\sharp(\tau)$ is similarly defined. $P(G, D) = P(G)P(D|G)$ is referred to as the *BD score* of structure G given data D.

Denoting as vector, $\theta_{x|\tau} = (\theta_{x_1|\tau}, ..., \theta_{x_s|\tau})$, the *Bayes estimate* of $\theta_{x|\tau}$ given D is

$$\theta^{be}_{x|\tau} = (\theta^{be}_{x_1|\tau}, \theta^{be}_{x_2|\tau}, ...) = \int \theta_{x|\tau} \ . \ \rho(\theta_{x|\tau}|D) d\theta_{x|\tau}. \tag{2.2}$$

We denote $\theta^{be}_{x|\tau} = \{\theta^{be}_{\chi|\tau}\}$ and $\theta^{be} = \{\theta^{be}_{x|\tau}\}$.

2.2. NAT-modeled BNs

A NAT model [6, 7] is defined over an effect e and a set of $n \geq 2$ causes $C = \{c_1, ..., c_n\}$, where $e \in D_e = \{e_0, ..., e_\eta\}$ $(\eta \geq 1)$ and $c_i \in \{c_{i0}, ..., c_{im_i}\}$ $(i = 1, ..., n;\ m_i \geq 1)$. C and e form a family in BN, whose dependence is quantified by CPDs in tabular BNs. Values e_0 and c_{i0} are *inactive*. Other values (may be written as e_+ or c_{i+}) are *active*.

For simplicity, we denote domain knowledge by $P()$ in this subsection (rather than by $Pr()$). A causal event is a *success* or *failure* depending on if e is active up to a given value, is *single-* or *multi-causal* depending on the number of active causes, and is *simple* or *congregate* depending on value range of e. $P(e_k \leftarrow c_{ij}) = P(e_k|c_{ij}, c_{z0} : \forall z \neq i)$ $(j > 0)$ is probability of a *simple single-causal success*, and

$$P(e \geq e_k \leftarrow c_{1j_1}, ..., c_{qj_q}) = P(e \geq e_k|c_{1j_1}, ..., c_{qj_q}, c_{z0} : c_z \in C \setminus X)$$

is probability of a *congregate multi-causal success*, where $j_1, ..., j_q > 0$, $X = \{c_1, ..., c_q\}$ $(q > 1)$. The latter may be denoted as $P(e \geq e_k \leftarrow \underline{x}_+)$. Interactions among causes may be reinforcing or undermining as defined below.

Definition 1. *Let e_k be an active effect value, $R = \{W_1, ..., W_m\}$ $(m \geq 2)$ be a given partition of a set $X \subseteq C$ of causes, $S \subset R$, and $Y = \cup_{W_i \in S} W_i$. Sets of causes in R reinforce each other relative to e_k, iff $\forall S$ $P(e \geq e_k \leftarrow \underline{y}_+) \leq P(e \geq e_k \leftarrow \underline{x}_+)$. They undermine each other iff $\forall S$ $P(e \geq e_k \leftarrow \underline{y}_+) > P(e \geq e_k \leftarrow \underline{x}_+)$.*

A NAT consists of one or more Non-Impeding Noisy-AND (NIN-AND) gates. A *direct* gate involves disjoint sets of causes $W_1, ..., W_m$. Each input event is a success $e \geq e_k \leftarrow \underline{w}_{i+}$ $(i = 1, ..., m)$, e.g., Fig. 2 (a) where W_i is a singleton. Output event $e \geq e_k \leftarrow \underline{w}_{1+}, ..., \underline{w}_{m+}$ has probability $\prod_{i=1}^{m} P(e \geq e_k \leftarrow \underline{w}_{i+})$. Direct gates encode undermining interactions. Each input of a *dual* gate is a failure $e < e_k \leftarrow \underline{w}_{i+}$, e.g., Fig. 2 (b). A dual gate output event $e < e_k \leftarrow \underline{w}_{1+}, ..., \underline{w}_{m+}$ has probability $\prod_{i=1}^{m} P(e < e_k \leftarrow \underline{w}_{i+})$ and satisfies relation $P(e < e_k \leftarrow ...) = 1 - P(e \geq e_k \leftarrow ...)$. Dual gates encode reinforcing interactions.

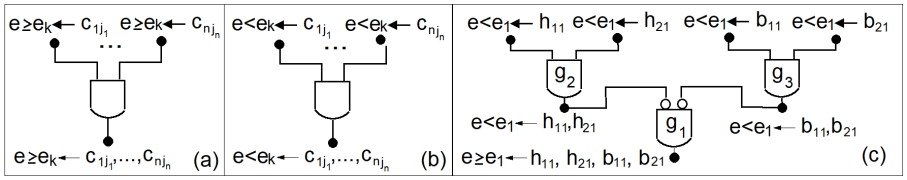

Figure 2. Direct gate (a), dual gate (b), and NAT (c).

Fig. 2 (c) shows a NAT, where causes h_1 and h_2 reinforce each other, and so do b_1 and b_2. However, the two groups undermine each other. That is, for gate g_1, each W_i (as in Def. 1) is a general set. See [6] for a formal definition of NATs. From the NAT and probabilities of its input events, in the general form $P(e_k \leftarrow c_{ij})$ $(j, k > 0)$, called *single-causals*, $P(e \geq e_1 \leftarrow h_{11}, h_{21}, b_{11}, b_{21})$ can be obtained. From single-causals and all

derivable NATs, CPDs $P(e|h_1, h_2, b_1, b_2)$ are uniquely specified. A NAT model is specified by the topology and single-causals with the space linear in n.

The *leaky* cause for an effect e represents all causes of e not explicitly named. A leaky cause may or may not be persistent [1]. A *non-persistent* leaky cause can be modeled the same way as other causes. A *persistent* leaky cause is always active and leads to special issues [7].

A BN where dependencies of some families are specified as NAT models (instead of tabular CPDs) is a *NAT-modeled BN*. If all families of more than one parent are NAT-modeled, the BN is *fully NAT-modeled*. A tabular BN has $O(Ks^n)$ space, while a fully NAT-modeled BN has $O(K\ s\ n)$ space. CPDs of a BN family can be approximated into a NAT model by *compression* [7]. Hence, a tabular BN can be approximated by a fully NAT-modeled BN. Inference methods for tabular BNs can be applied to NAT-modeled BNs by converting them into efficient tabular BNs, e.g., by trans-causalization [14]. The inference is tractable when NAT-modeled BNs have high treewidth and low density.

3. Learning NAT-model BN Structure with BD Scores

Learning structures of tabular BNs from data has been actively researched since 1990s. A widely applied approach is to evaluate each candidate structure by a scoring function, such as MDL or BD, and to limit the exponential structure space by heuristic search. To overcome the limitation of tabular BNs considered in Section 1, learning BNs with local models has also been pursued [15–17]. Work in [18, 19] explored local equality conditions such as CSI with decision trees or decision graphs as local structures, based on MDL or BD scores. Inequality conditions such as those in Def. 1 were explored in learning NAT-modeled BNs based on MDL score [13].

In this work, we present the first study of structure learning of NAT-modeled BNs based on the Bayesian approach and BD score. A NAT-modeled BN consists of a global DAG structure G, a local NAT structure L (including NAT topologies for all NAT-modeled families), single-causal parameters for all *NAT families*, and CPD parameters of the remaining *tabular families*. Given data D, we evaluate a candidate structure (G, L) by BD score

$$P(G, L|D) = \alpha P(D|G, L)P(L|G)P(G),$$

where α is the normalizing constant $1/P(D)$. In the following sections, we consider each of the 3 components, $P(D|G, L)$, $P(L|G)$, and $P(G)$, which we refer to as *likelihood*, *local structure prior*, and *global structure prior*.

Note that the learned BN is not necessarily fully NAT-modeled: Whether a family in the outcome structure is NAT-modeled or tabular depends on the score and search.

4. Likelihood

To define likelihood $P(D|G, L)$ for a NAT-modeled structure (G, L), we extend the meta-net for learning tabular BNs to learning NAT-modeled BNs by representing local NAT models and single-causal parameters. We do so with an example first and then generalize.

[**NAT-modeled meta-nets**] Consider the base-net G in Fig. 3 (a), where $V = \{a, b, c, d\}$, all variables are binary, and data D has size $N = 2$. Since c has 2 parents in G, L may specify its family to be tabular (*tab*) or NAT-modeled. If NAT-modeled, it may be a direct NIN-AND gate (*di*) or a dual gate (*du*). This local model type is represented in the meta-net by variable $\omega_c \in \{tab, di, du\}$ in Fig. 3 (b). Since (G, L) is given, $P(\omega_c)$ consists of extreme values. For instance, if L specifies c family as a direct gate, then $P(\omega_c = di) = 1$.

If family of c is tabular, it has 4 CPDs, and the meta-net has 4 corresponding θ nodes (Fig. 3 (b) bottom). If family of c is NAT-modeled, only two θ nodes are well-defined:

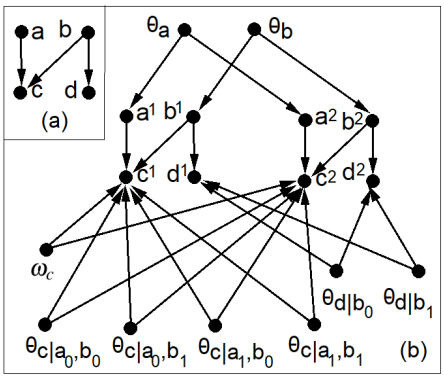

Figure 3. (a) Base-net structure G. (b) Meta-net.

$\theta_{c \leftarrow a_1} = \theta_{c|a_1,b_0}$ and $\theta_{c \leftarrow b_1} = \theta_{c|a_0,b_1}$. Making ω_c as a parent of c allows all such cases to be handled correctly through CPD $P(c|a, b, \omega_c, \theta_{c|a_0,b_0}, \theta_{c|a_0,b_1}, \theta_{c|a_1,b_0}, \theta_{c|a_1,b_1})$.

In general, for each variable x of 2 or more parents in G, the meta-net has a ω_x variable on its local model type. The domain of ω_x includes value *tab* for tabular, and possible NATs for the x family. For instance, if x has 5 parents in G, its family has 472 possible NAT models, and ω_x has domain size 473. Given (G, L), we have $P(\omega_x) \in \{0, 1\}$.

For each $x \in V$, the meta-net contains as many θ nodes as the meta-net for tabular BNs, one per CPD $Pr(x|\pi = \tau)$. When x family is NAT-modeled, each necessary θ node maps to a single-causal distribution $Pr(x \leftarrow c_{ij})$, where c_{ij} is an active value of cause c_i, and remaining θ nodes are superfluous. Since $Pr(x \leftarrow c_{ij}) = Pr(x|\tau)$, where τ has exactly one active value, we denote the θ node by $\theta_{x|\tau}$ (instead of $\theta_{x \leftarrow c_{ij}}$) for consistency with the tabular case. Hence, when local structure L asserts x family to be NAT-modeled, only single-causal θ nodes are relevant. This dynamic dependency is effected through CPD $P(x|\pi, \omega_x, ...)$ in the meta-net. Each θ node has a Dirichlet prior pdf.

[Properties of NAT-modeled meta-nets] We refer to meta-nets for learning tabular BNs as *T-meta-nets*. We term meta-nets for learning NAT-modeled BNs (defined above) as *N-meta-nets*. N-meta-nets have the following properties.

Theorem 1. *Every NAT-modeled BN structure (G, L) over V has a well-defined N-meta-net given complete data D.*

Proof: Let (G, L) be a NAT-modeled BN structure and D be the data over V. Initialize the N-meta-net as the T-meta-net for a tabular BN of structure G and data D, which consists of one instance of G for each record of D, a θ node for each CPD of the tabular BN, and corresponding arcs. For each variable $v \in V$, denote the set of θ nodes, one per CPD of v, by Θ_v.

For each $x \in V$ of parents π in G, where $|\pi| \geq 2$, add a variable ω_x to the N-meta-net. The domain of ω_x consists of value *tab* and one value for each possible NAT topology over the x family. For each copy of x (one per record of D) in the N-meta-net, add ω_x as a parent, set CPDs $P(x|\pi, \omega_x, \Theta_x)$, and set $P(\omega_x)$ to be deterministic according to the local model type of x specified by L. The N-meta-net for (G, L) and D is now constructed. [End]

Note that existence of ω variables allows an N-meta-net to easily switch among all alternative NAT topologies for each NAT family. In other words, the N-meta-net can easily switch among all L local structures for a given global structure G, by modifying the prior distributions of ω variables. Furthermore, N-meta-nets are used to derive BD scores for NAT-modeled BNs, but are not directly computed during structure learning, as seen below.

Next, we show that *parameter independence* of T-meta-nets (see, e.g., [11]) also applies to N-meta-nets. The proof utilizes the well-known d-separation [20].

Theorem 2. *In an N-meta-net, any two disjoint subsets of θ variables are independent.*

Proof: It suffices to show that any two θ nodes are d-separated. Each θ node has only outgoing arcs in the N-meta-net. Hence, any path between two θ nodes u and v cannot be directed. There must be a node x that is head-to-head on the path $(\rightarrow x \leftarrow)$, which blocks the path, e.g., the path $(\theta_a, a^1, c^1, b^1, \theta_b)$ in Fig. 3 (b). Since every such path is blocked, u and v are d-separated. [End]

Parameter independence of N-meta-nets also holds conditioned on data D:

Theorem 3. *In an N-meta-net, any two disjoint subsets of θ variables are independent conditioned on a complete data set D.*

Proof: It suffices to show that any two θ nodes are d-separated conditioned on D. Consider a path between θ nodes u and v, with remaining nodes on the path $Z \subset V$. Since D is complete, every $z \in Z$ is observed. If there is a node z that is head-to-tail $(\rightarrow z \rightarrow)$ or tail-to-tail $(\leftarrow z \rightarrow)$ on the path, then the path is blocked, e.g., $(\theta_a, a^1, c^1, b^1, d^1, \theta_{d|b_0})$ in Fig. 3 (b).

If no node $z \in Z$ exists on the path that is head-to-tail or tail-to-tail, then the path must be $u \rightarrow x \leftarrow v$, where x is head-to-head. It must be the case that u denotes $\theta_{x|\tau}$, v denotes $\theta_{x|\tau'}$, and instantiations τ and τ' of parents π of x (in G) differ, e.g., $(\theta_{c|a_0,b_0}, c^1, \theta_{c|a_0,b_1})$ in Fig. 3 (b). Since at most one of τ and τ' is consistent with D, at least one arc, $u \rightarrow x$ or $x \leftarrow v$, can be equivalently removed, and the path disappears.

Since every path between u and v is either blocked or can be equivalently removed, u and v are d-separated. [End]

Note that Theorem 2 still holds if the subsets include ω variables. However, that is not the case for Theorem 3. Conditioned on D, a θ node and a related ω node are not d-separated. For instance, the path $(\omega_c, c^1, \theta_{c|a_0,b_1})$ in Fig. 3 (b) is not blocked: Depending on the local model type, the θ node plays different roles.

[Subscore of tabular family] Consider likelihood score $P(D|G, L)$ next. By Eqn. (2.1), $P(D|G)$ for tabular BNs is decomposable by (x, π) family, as well as by $\pi = \tau$:

$$P(D|G) = \prod_{x \in V} SS(D, x) = \prod_{x \in V} \left(\prod_{\tau} SS(D, x, \tau) \right),$$

where subscore for family (x, π) with $\pi = \tau$ is

$$SS(D, x, \tau) = \frac{\Gamma(\psi_{x|\tau})}{\Gamma(\psi_{x|\tau} + \sharp(\tau))} \prod_{\chi} \frac{\Gamma(\psi_{\chi|\tau} + \sharp(\chi, \tau))}{\Gamma(\psi_{\chi|\tau})}, \tag{4.1}$$

and subscore for family (x, π) is

$$SS(D, x) = \prod_{\tau} SS(D, x, \tau). \tag{4.2}$$

The decomposability is a direct consequence of parameter independence in T-meta-nets. By Theorem 3 on parameter independence in N-meta-nets, we have decomposability for NAT-modeled likelihood $P(D|G, L)$:

$$P(D|G, L) = \prod_{x \in V} SS(D, x). \tag{4.3}$$

As N-meta-nets represent tabular families equivalently as T-meta-nets, subscore $SS(D, x)$ for a tabular (x, π) family can be evaluated by Eqns. (4.2) and (4.1).

[**Extending Bayes estimates of parameters**] BD scores for tabular BNs can be derived using the following relation [11]:

$$P(\alpha|G,D) = Pr_{\theta^{be}}(\alpha), \tag{4.4}$$

where α is an event over V, $P(\alpha|G,D)$ is its posterior from the T-meta-net, and $Pr_{\theta^{be}}(\alpha)$ is obtained from the base-net parameterized by Bayes estimates. In N-meta-nets, only single-causal θ nodes are well-defined for NAT families. Non-single-causal parameters are undefined (their θ nodes are superfluous), and so are their Bayes estimates, which limits applicability of the above relation. To resolve this issue, we extend Bayes estimates to count for NAT-models:

Definition 2. *Let (x,π) be a NAT-modeled family. Instantiation $\pi=\tau$ is* **Type 0** *if τ has no active value, is* **Type 1** *if τ has exactly 1 active value, and is* **Type 2** *if τ has 2 or more active values.*

If τ is Type 1, Bayes estimate $\theta^{be}_{x|\tau}$ follows Eqn. (2.2). If τ is Type 2, $\theta^{be}_{x|\tau}$ is deterministically derived from Type 1 Bayes estimates. If τ is Type 0, $\theta^{be}_{x|\tau} \in \{0,1\}$.

In the N-meta-net of Fig. 3, if $\omega_c = di$, $\theta_{c|a_0,b_1}$ and $\theta_{c|a_1,b_0}$ are well-defined, but $\theta_{c|a_0,b_0}$ and $\theta_{c|a_1,b_1}$ are superfluous. When $\theta^{be}_{c|a_0,b_1} = (0.1, 0.9)$ and $\theta^{be}_{c|a_1,b_0} = (0.3, 0.7)$ (τ is Type 1), we have $\theta^{be}_{c|a_0,b_0} = (1, 0)$ (τ is Type 0) and $\theta^{be}_{c|a_1,b_1} = (0.37, 0.63)$ (τ is Type 2).

[**Subscore of NAT family**] From Def. 2 and Eqn. (4.4), we have Eqn. (4.5) for NAT-modeled BNs:

$$P(\alpha|G,L,D) = Pr_{\theta^{be}}(\alpha), \tag{4.5}$$

where $P(\alpha|G,L,D)$ is from the N-meta-net. Denote data of size N by $D = (d^1, ..., d^N)$ and $D_i = (d^1, ..., d^i)$. From Eqn. (4.5), we have $P(d^i|G,L,D_{i-1}) = Pr_{\theta^{be}_{i-1}}(d^i)$, where θ^{be}_{i-1} is Bayes estimates given D_{i-1}. By the chain rule of BNs, $Pr_{\theta^{be}_{i-1}}(d^i) = \prod_{\chi,\tau\sim d^i} \theta^{be}_{\chi|\tau,i-1}$, where $\chi,\tau \sim d^i$ selects (x,π) that is consistent with d^i. From the above, we have

$$P(D|G,L) = \prod_{i=1}^{N} P(d^i|G,L,D_{i-1}) = \prod_{i=1}^{N} \prod_{\chi,\tau\sim d^i} \theta^{be}_{\chi|\tau,i-1} = \prod_{x\in V} \left(\prod_{i=1}^{N} \theta^{be}_{\substack{\chi|\tau,i-1 \\ \chi,\tau\sim d^i}} \right) = \prod_{x\in V} SS(D,x).$$

Since D is *exchangeable*, the order of data records in the 2nd expression does not matter. The 4th expression means that factors for each x family are independent of others. Hence, in 4th expression, data records relative to each x may be ordered differently. Therefore, for each NAT x family, we order records in the 4th expression from 1 to N by type of τ: Type 0 records first, followed by Type 2, and followed by Type 3, breaking ties arbitrarily. We analyze subscore $SS(D,x)$ for a NAT family below, using this order:

Consider contribution of record d^i to $SS(D,x)$, where $\chi,\tau \sim d^i$ and τ is Type 0. If χ is active, (χ,τ) is impossible for NAT x family. Hence, $\theta^{be}_{\chi|\tau,i-1} = 0$, $SS(D,x) = 0$, and $P(D|G,L) = 0$. It signifies that (x,π) being NAT contradicts data D. Hence, either (x,π) is tabular, or (x,π) is NAT with an (extra) persistent leaky cause. On the other hand, if χ is inactive, then

$$\theta^{be}_{\chi|\tau,i-1} = 1, \tag{4.6}$$

without visible impact to $SS(D,x)$. In summary, if D has any d^i where χ is active and τ is Type 0, $P(D|G,L) = 0$. If χ is inactive, d^i can be ignored when processing $SS(D,x)$.

If τ is Type 1, contribution of record d^i to $SS(D,x)$ is

$$\theta^{be}_{\chi|\tau,i-1} = \frac{\psi_{\chi|\tau} + \sharp_{i-1}(\chi,\tau)}{\psi_{x|\tau} + \sharp_{i-1}(\tau)}, \tag{4.7}$$

where $\sharp_{i-1}(\chi,\tau)$ counts records in D_{i-1} that instantiate family of x to $x=\chi$ and $\pi=\tau$. The collective contribution of all records where $\chi,\tau \sim d^i$ is $SS(D,x,\tau)$ by Eqn. (4.1).

If τ is Type 2, contribution of record d^i to $SS(D,x)$ is $\theta^{be}_{\chi|\tau,i-1}$, where $\chi,\tau \sim d^i$, and it is determined by the x family NAT and relevant Type 1 Bayes estimates $\theta^{be}_{\chi'|\tau',i-1}$, each according to Eqn. (4.7). Due to the above type-based record ordering, all Type 1 records are indexed lower than d^i. Hence,

$$\theta^{be}_{\chi'|\tau',i-1} = \frac{\psi_{\chi'|\tau'} + \sharp_{i-1}(\chi',\tau')}{\psi_{x|\tau'} + \sharp_{i-1}(\tau')} = \frac{\psi_{\chi'|\tau'} + \sharp(\chi',\tau')}{\psi_{x|\tau'} + \sharp(\tau')} = \theta^{be}_{\chi'|\tau'}.$$

It then follows that contribution of d^i to $SS(D,x)$ is independent of index i:

$$\theta^{be}_{\chi|\tau,i-1} = \theta^{be}_{\chi|\tau}. \tag{4.8}$$

By Eqn. (4.8), if $(\chi|\tau)$ occurs m times in D, they contribute $(\theta^{be}_{\chi|\tau})^m$ to $SS(D,x)$. This reveals that the type-based record ordering is only convenient for justifying soundness, but is not algorithmically necessary.

For example, consider G in Fig. 3 (a), L with c family modeled as a direct gate NAT, and data D in Table 1. Families of a,b,d are tabular, and their $SS(D,x)$ are 1/72, 1/252, 1/630, respectively. For NAT c family, numbers of records for Type 0, 1, 2 in D are 1, 5, 2, respectively. Type 0 record has no visible impact to $P(D|G,L)$. Type 1 records contribute 1/60 to $SS(D,c)$, and produce $\theta^{be}_{c_1|a_1,b_0} = 3/7$ and $\theta^{be}_{c_1|a_0,b_1} = 1/2$. Hence, $\theta^{be}_{c_1|a_1,b_1} = 3/14$ and Type 2 records contribute 9/196 to $SS(D,c)$. We have $SS(D,c) = 3/3920$ and $P(D|G,L) = 6.695 \times 10^{-11}$.

Table 1. Example data D

a	b	c	d
a_0	b_0	c_0	d_1
a_1	b_0	c_0	d_1
a_1	b_0	c_1	d_0
a_1	b_0	c_0	d_0
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_0
a_1	b_0	c_1	d_0
a_1	b_0	c_0	d_0

5. **Local Structure Prior**

We next consider local structure prior $P(L|G)$, where L specifies the local structure for every x family in G. Learning BNs with local decision trees was studied in [18] based on MDL scores, and an alternative BD score was proposed with

$$P(L|G) = \alpha 2^{-DL(L)}, \tag{5.1}$$

where $DL()$ is description length under MDL principle, and α is normalizing constant. Applying the idea to NAT-modeled BNs, we specify $DL(x,L_x)$ for each x family, where L_x extracts local structure for x family from L, and

$$DL(L) = \sum_{x\in V} DL(x,L_x). \tag{5.2}$$

If x family is tabular by L, we have (see [13])

$$DL(x,L_x = tab) = \frac{1}{2} log_2(N)|Pr(x|\pi)|. \tag{5.3}$$

If x family is structured as a NAT T_x, we have

$$DL(x,L_x = nat) = DL(T_x) + DL(SC_x), \tag{5.4}$$

where $DL(T_x)$ and $DL(SC_x)$ are description lengths of NAT and single-causals [13].

For example, consider G in Fig. 3 (a), L with c family NAT-modeled, and data size $N = 8$. We have DL for a, b, c, d as 1.5, 1.5, 4, 3, respectively, and the local structure prior $P(L|G)$ without α is $0.3535 * 0.3535 * 0.0625 * 0.125 = 0.000976$.

6. Global Structure Prior

We consider global structure prior $P(G)$. Preference of simpler DAG G is suggested in [8, 15] by assuming (1) independent parent sets:

$$P(G) = \prod_{x \in V} P(\pi), \tag{6.1}$$

and (2) independent individual parents: $P(y \to x | z \to x) = P(y \to x)$, where $y \to x$ is an arc in G. Since no specific form of $P(y \to x)$ is suggested in [8, 15], we develop the following:

We assign

$$P(\pi) = \eta \; k^{|\pi|}, \tag{6.2}$$

where $k \in (0, 1)$ and η is a constant. When x is root in G, $P(\pi = \emptyset) = \eta$. It can be shown that the assignment satisfies the following properties, which favor simpler structures:

(1) If x has w parents and v has $q > w$ parents, then $P(\pi_x) > P(\pi_v)$.
(2) If G has n nodes, then $P(G) = \eta^n k^m$, where m counts the number of arcs in G.

From the 2nd property, constant η can be ignored when comparing two DAGs. This is desirable as the number of alternative G is intractable. As an example, for G in Fig. 3 (a), assuming $k = 0.5$, the global structure prior without η is $1 * 1 * 0.25 * 0.5 = 0.125$.

We conclude BD scores for learning NAT-modeled BNs with their decomposability:

Theorem 4. *The likelihood, local structure prior, and global structure prior defined above for learning NAT-modeled BNs are each decomposable by variable family, and so is the BD score specified by them.*

Proof: From Eqns. (4.3), (4.2), (4.1), (4.6), (4.7), and (4.8), it follows that $P(D|G, L)$ is multiplicatively decomposable by $x \in V$. From Eqns. (5.1), (5.2), (5.3), and (5.4), $P(L|G)$ is multiplicatively decomposable by $x \in V$. From Eqns. (6.1) and (6.2), $P(G)$ is multiplicatively decomposable by $x \in V$. The theorem then follows. [End]

7. Algorithm and Complexity

Learning BNs is NP-complete, and learning NAT-modeled BNs involves an even larger (G, L) space. For instance, a single NAT family of 8 causes has 1,320,064 alternative NAT structures (each encodes a unique set of causal interactions among the causes). A given G with exactly two such NAT families would have 1.7×10^{12} alternative L structures. To improve efficiency of learning, we apply heuristic search as presented below. The presentation focuses on structure learning, although our implementation (Section 8) also learns parameters (tabular CPDs and NAT single-causals).

The top level algorithm *LearnNatBnByBD* takes data D over V as input. It learns a NAT-modeled BN structure (G, L), where G is possibly over a superset of V (due to persistent leaky causes). It uses heuristic search to find a best structure over the intractable (G, L) space.

It adopts a sequence of (G, L), from empty G to the final structure. Each (G, L) is computed by *OneRoundSearch* and improves BD score. It differs from the previous (G, L) by one arc (through arc operation *add*, *delete* or *reverse*), and may change local structure for up to two families.

Algorithm 1. *LearnNatBnByBD*(D, V)
1 init (G, L) *to empty DAG, init Score to BD score of* (G, L)*, Done = false;*
2 while Done = false,
3 Done = true;
4 $(G', L', Score') = OneRoundSearch(G, L, V, Score, D);$
5 if $Score' > Score$*, then* $(G, L) = (G', L')$*,* $Score = Score'$*, Done = false;*
6 return (G, L);

For *OneRoundSearch*, G is the best DAG before the round, G' is a new DAG to be evaluated, G^* is the best new DAG so far in the round, and $Score^* \geq Score$ on return. It calls *getBDScore* for its key computation.

Algorithm 2. *OneRoundSearch*$(G, L, V, Score, D)$
1 $(G^*, L^*) = (G, L)$*,* $Score^* = Score;$
2 for each pair of $u, v \in V$*,*
3 for each valid arc operation Op on G^* *over* u, v*,*
4 apply Op on G^* *to obtain* G'*,*
5 if G' *is cyclic, continue;*
6 $(L', Score') = getBDScore(G^*, L^*, Score^*, G', D);$
7 if $Score' > Score^*$*, then* $(G^*, L^*) = (G', L')$*,* $Score^* = Score';$
8 return $(G^*, L^*, Score^*);$

As input to *getBDScore* (see below), G and G' differ by one arc. By Theorem 4, it suffices for *getBDScore* to evaluate only subscores for families modified in G' (lines 3 and 8), and update $Score'$ (line 9). To avoid evaluating intractably many NATs for any x family, the best NAT is selected heuristically by *compression* [7] (lines 6 and 7), rather than by direct score evaluation. In line 6, x_0 denotes inactive value of x and π_0 denotes inactive instantiation of π. On return from *getBDScore*, L' differs from L by up to two families and $Score' \geq Score$.

Algorithm 3. *getBDScore*$(G, L, Score, G', D)$
1 $L' = L;$ $Score' = Score;$
2 for each $x \in V$ *whose family structure in* G' *differs from* G*,*
3 get tabular subscore SS_{tab} *for* x *family from* G'*,* D *by Eqns. (4.1), (4.2), (5.3), (6.2);*
4 for parents π *of* x *in* G'*, if* $|\pi| \geq 2$*,*
5 estimate $Pr(x|\pi)$ *from* D*;*
6 if $Pr(x_0|\pi_0) = 1$*, compress* $Pr(x|\pi)$ *to get NAT* T_x*;*
7 else compress $Pr(x|\pi)$ *to get NAT* T_x *with persistent leaky cause;*
8 get NAT subscore SS_{nat} *for* x *family from* G'*,* D *by Eqns. (4.7), (4.8), (5.4), (6.2);*
9 based on comparison of SS_{tab} *and* SS_{nat}*, update Score' and* L' *on* x *family;*
10 return $(L', Score');$

NAT T_x (lines 6 and 7) is obtained from $Pr(x|\pi)$. However, single-causals obtained by compression for T_x may differ from values in $Pr(x|\pi)$. This occurs when dependency embedded in $Pr(x|\pi)$ differs from any NAT model. On the other hand, SS_{nat} in line 8 is based on single-causals from $Pr(x|\pi)$. Hence, whenever $Pr(x|\pi)$ differs significantly from any NAT model, $SS_{nat} < SS_{tab}$ is expected: resulting in rejecting NAT model for x family.

For complexity of *LearnNatBnByBD*, denote $K = |V|$. *OneRoundSearch* evaluates $O(K^2)$ arcs before one is added, removed, or reversed. It adds at most one arc, and at most $O(K^2)$ arcs may be added. Each arc cannot be repeatedly added, reversed, or deleted, and also improve *Score*. Hence, the number of executions of *OneRoundSearch* is $O(K^2)$, and complexity of *LearnNatBnByBD* is $O(K^4)$.

The NAT-model compression during *getBDScore* involves a significant cost, whose complexity [7] is left implicit in the above analysis, and must be counted for. Before running

LearnNatBnByBD, D is pre-processed into a set F of frequencies of unique records. The complexity of *LearnNatBnByBD* is linear on $|F|$ and $|F| << |D|$.

8. Experimental Study

Preliminary experiment is conducted to evaluate the above BD score and structure learning algorithm with the objective below: Suppose that the data-generating environment can be expressed as a NAT-modeled BN B_1. Then an equivalent tabular BN B_2 exists with its joint probability distribution (JPD) identical to B_1. We want to answer the question: Can our algorithm learn a NAT-modeled BN B_3 such that posterior marginals by inference with B_3 approximate those by B_2 well, and at the same time inference with B_3 is significantly more efficient than B_2?

We simulated 30 fully NAT-modeled *source* BNs (B_1), each of 80 binary or ternary variables. Each variable has a maximum of 12 parents. The DAG of each source BN has 5% extra arcs beyond being singly connected. Hence, each source BN is multiply connected with a high treewidth (at least 12) and low density. For each source BN, the equivalent tabular *peer* BN (B_2) is derived, from which a data set of size 5000 is sampled. *LearnNatBnByBD* is implemented in Java and run on a desktop with AMD Ryzen 7 5800X 8-Core processor at 3.8 GHz by single-thread computation, to learn a NAT-modeled BN (B_3) from each data set.

For each pair of peer BN and learned BN, we performed 10 runs of inferences, each with random observations on 2 to 8 variables (up to 10% of all variables): a total of 600 runs. Posterior marginal errors of learned BNs, relative to peer BNs, averaged over 10 runs are shown in Fig. 4 (a). Posterior marginals from learned BNs approximate those from peer BNs sufficiently well (average error at about 0.025).

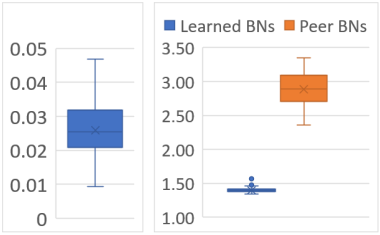

Figure 4. (a) Inference errors. (b) Runtime.

As shown in Fig. 4 (b), learned BNs have average inference runtime (msec in log10) of 25 msec, while peer BNs take about 900 msec. Hence, learned BNs are significantly more efficient than peer BNs (36 times faster on average).

9. Conclusion

We presented the first Bayesian framework for learning structures of BNs with NAT local models, where NAT models are chosen due to their multiple merits. In particular, we extended meta-nets for learning tabular BNs to N-meta-nets to enable representation of NAT-modeled families and single-causal parameters. We showed formally that N-meta-nets are expressive, and satisfy parameter independence. Using N-meta-nets, we developed BD scores for learning NAT-modeled BN structures, consisting of likelihood, local structure prior and global structure prior. The BD scores were shown to be decomposable. A heuristic algorithm for learning NAT-modeled BN structures with BD scores is presented for search through the structure space that is significantly more complex than the space in learning

tabular BNs. Our experiment showed that when the data-generating environment can be expressed as NAT-modeled BNs, a NAT-modeled BN can be learned whose inference is sufficiently accurate while being significantly more efficient than the tabular BN alternative.

Acknowledgements

Financial support from NSERC Discovery Grant to the first author is acknowledged.

References

[1] M. Henrion. "Some practical issues in constructing belief networks". In: *Uncertainty in Artificial Intelligence 3*. Ed. by L. Kanal, T. Levitt, and J. Lemmer. Elsevier Science Publishers, 1989, pp. 161–173.

[2] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. "Context-specific independence in Bayesian networks". In: *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*. 1996, pp. 115–123.

[3] P. Maaskant and M. Druzdzel. "An Independence of Causal Interactions Model for Opposing Influences". In: *Proc. 4th European Workshop on Probabilistic Graphical Models*. Ed. by M. Jaeger and T. Nielsen. Hirtshals, Denmark, 2008, pp. 185–192.

[4] J. Vomlel and P. Tichavsky. "An approximate tensor-based inference method applied to the game of Minesweeper". In: *Proc. 7th European Workshop on Probabilistic Graphical Models, Springer LNAI 8745*. 2012, pp. 535–550.

[5] S. Woudenberg, L. van der Gaag, and C. Rademaker. "An intercausal cancellation model for Bayesian-network engineering". In: *Inter. J. Approximate Reasoning* 63 (2015), pp. 32–47.

[6] Y. Xiang. "Non-impeding Noisy-AND Tree Causal Models Over Multi-valued Variables". In: *International J. Approximate Reasoning* 53.7 (2012), pp. 988–1002.

[7] Y. Xiang and Q. Jiang. "NAT Model Based Compression of Bayesian Network CPTs over Multi-Valued Variables". In: *Computational Intelligence* 34.1 (2018), pp. 219–240.

[8] G. Cooper and E. Herskovits. "A Bayesian method for the induction of probabilistic networks from data". In: *Machine Learning* 9 (1992), pp. 309–347.

[9] W. Lam and F. Bacchus. "Learning Bayesian networks: an approach based on the MDL principle". In: *Computational Intelligence* 10.3 (1994), pp. 269–293.

[10] R. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2004.

[11] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge Univ Press, 2009.

[12] X. Zheng and E. X. B. Aragam P. Ravikumar. "Dags with no tears: continuous optimization for structure learning". In: *Advances in Neural Information Processing Systems 31*. 2018.

[13] Y. Xiang and Q. Wang. "Learning Tractable NAT-Modeled Bayesian Networks". In: *Annals of Mathematics and Artificial Intelligence, DOI: 10.1007/s10472-021-09748-0* (2021).

[14] Y. Xiang and D. Loker. "Trans-Causalizing NAT-Modeled Bayesian Networks". In: *IEEE Trans. Cybernetics, DOI: 10.1109/TCYB.2020.3009929* (2020).

[15] W. Buntine. "Theory refinement on Bayesian networks". In: *Proc. 7th Conf. on Uncertainty in Artificial Intelligence*. 1991, pp. 52–60.

[16] F. Diez. "Parameter adjustment in Bayes networks: The generalized noisy OR-gate". In: *Proc. 9th Conf. on Uncertainty in Artificial Intelligence*. Ed. by D. Heckerman and A. Mamdani. Morgan Kaufmann, 1993, pp. 99–105.

[17] C. Meek and D. Heckerman. "Structure and parameter learning for causal independence and causal interaction models". In: *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*. 1997, pp. 366–375.

[18] N. Friedman and M. Goldszmidt. "Learning Bayesian networks with local structure". In: *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1996, pp. 252–262.

[19] D. Chickering, D. Heckerman, and C. Meek. "A Bayesian approach to learning Bayesian networks with local structure". In: *Proc. of 13th Conf. on Uncertainty in Artificial Intelligence*. 1997, pp. 80–89.

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