Conditional Independence in Testing Bayesian Networks

Yujia Shen [1] **Haiying Huang** [1] **Arthur Choi** [1] **Adnan Darwiche** [1]

Abstract

Testing Bayesian Networks (TBNs) were introduced recently to represent a set of distributions, one of which is selected based on the given evidence and used for reasoning. TBNs are more expressive than classical Bayesian Networks (BNs): Marginal queries correspond to multi-linear functions in BNs and to piecewise multi-linear functions in TBNs. Moreover, TBN queries are universal approximators, like neural networks. In this paper, we study conditional independence in TBNs, showing that it can be inferred from d-separation as in BNs. We also study the role of TBN expressiveness and independence in dealing with the problem of learning with incomplete models (i.e., ones that miss nodes or edges from the data-generating model). Finally, we illustrate our results on a number of concrete examples, including a case study on Hidden Markov Models.

1. Introduction

Testing Bayesian Networks (TBNs) were introduced recently, motivated by an expressiveness gap between Bayesian and neural networks (Choi & Darwiche, 2018). The basic observation here is that neural networks are universal approximators, which means that they can approximate any continuous function to an arbitrary error[1] (Hornik et al., 1989; Cybenko, 1989; Leshno et al., 1993). However, for Bayesian networks (BNs), a joint marginal query is a multi-linear function of evidence and a conditional marginal query corresponds to a quotient of multi-linear functions.

The main insight behind TBNs is that a TBN represents a set of distributions instead of just one. Moreover, one of these distributions is selected based on the given evidence and used for reasoning. As a result, in a TBN, a joint marginal query corresponds to a piecewise multi-linear function and a conditional marginal query corresponds to a quotient of such functions. TBNs were shown to be universal approximators in the following sense. Any continuous function can be approximated to an arbitrary error by a marginal query on a carefully crafted TBN (under similar assumptions to those used in neural networks). Therefore, as function approximators, TBNs are as expressive as neural networks. Moreover, TBNs are more expressive than BNs as they can capture some relations between evidence and marginal probabilities that cannot be captured by BNs.

We further investigate TBNs in this paper from several angles. First, we consider the notion of conditional independence, which is somewhat subtle in TBNs since the addition of evidence can change the distribution selected by a TBN for reasoning. In particular, we show that conditional independence can still be inferred from the structure of a TBN using the classical notion of d-separation despite this more dynamic behavior. Next, we consider some situations in discriminative learning where the expressiveness of TBNs provide an advantage. In particular, we consider learning with incomplete BNs, which miss some nodes or edges from the data-generating BN, showing analytically how TBNs can help alleviate this problem. Finally, we extend and further analyze the mechanism used by TBNs for selecting distributions based on evidence, which increases the reach of TBNs and tightens our understanding of their semantics.

This paper is structured as follows. We review TBNs in Section 2 and then extend their dependence on evidence in Section 3. We study conditional independence in Section 4, proving it can be inferred from d-separation as in BNs. We then consider discriminative learning in Section 5, showing how TBNs can help alleviate the problem of learning with incomplete models. We follow by a case study in Section 6 on Hidden Markov Models and the associated problem of missing temporal dependencies. We finally close with some concluding remarks in Section 7. Proofs of results are delegated to Appendix in the supplementary material.

2. Testing Bayesian Networks

A Testing Bayesian Network (TBN) is a BN whose CPTs are selected dynamically based on the given evidence.

[1] Computer Science Department, University of California, Los Angeles, California, USA. Correspondence to: Yujia Shen <yujias@cs.ucla.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] Typically, the function is assumed to be defined on a compact set (i.e., closed and bounded) and hence uniformly continuous.

Consider a BN that contains a binary node X having a single binary parent U. The CPT for node X contains *one* distribution on X for each state u of its parent:

U	X	
u	x	$\theta_{x\|u}$
u	$\bar{x}$	$\theta_{\bar{x}\|u}$
$\bar{u}$	x	$\theta_{x\|\bar{u}}$
$\bar{u}$	$\bar{x}$	$\theta_{\bar{x}\|\bar{u}}$

In a TBN, node X can be *testing,* requiring *two* distributions on X for each state u of its parent, and a *threshold* for each state u, which is used to select one of these distributions:

U	X			
u	x	T_u	$\theta^+_{x\|u}$	$\theta^-_{x\|u}$
u	$\bar{x}$		$\theta^+_{\bar{x}\|u}$	$\theta^-_{\bar{x}\|u}$
$\bar{u}$	x	$T_{\bar{u}}$	$\theta^+_{x\|\bar{u}}$	$\theta^-_{x\|\bar{u}}$
$\bar{u}$	$\bar{x}$		$\theta^+_{\bar{x}\|\bar{u}}$	$\theta^-_{\bar{x}\|\bar{u}}$

The selection of distributions utilizes the posterior on parent U given some of the evidence on X's *non-descendants.*[2] For parent state u, the selected distribution on X is $(\theta^+_{x|u}, \theta^+_{\bar{x}|u})$ if the posterior on u is $\geq T_u$; otherwise, it is $(\theta^-_{x|u}, \theta^-_{\bar{x}|u})$. For parent state $\bar{u}$, the distribution is $(\theta^+_{x|\bar{u}}, \theta^+_{\bar{x}|\bar{u}})$ if the posterior on $\bar{u}$ is $\geq T_{\bar{u}}$; otherwise, it is $(\theta^-_{x|\bar{u}}, \theta^-_{\bar{x}|\bar{u}})$. Thus, the CPT for node X is determined *dynamically* based on the two thresholds and the posterior over parent U, leading to one of the following four CPTs:[3]

U	X	CPT_1	CPT_2	CPT_3	CPT_4
u	x	$\theta^+_{x\|u}$	$\theta^+_{x\|u}$	$\theta^-_{x\|u}$	$\theta^-_{x\|u}$
u	$\bar{x}$	$\theta^+_{\bar{x}\|u}$	$\theta^+_{\bar{x}\|u}$	$\theta^-_{\bar{x}\|u}$	$\theta^-_{\bar{x}\|u}$
$\bar{u}$	x	$\theta^+_{x\|\bar{u}}$	$\theta^-_{x\|\bar{u}}$	$\theta^+_{x\|\bar{u}}$	$\theta^-_{x\|\bar{u}}$
$\bar{u}$	$\bar{x}$	$\theta^+_{\bar{x}\|\bar{u}}$	$\theta^-_{\bar{x}\|\bar{u}}$	$\theta^+_{\bar{x}\|\bar{u}}$	$\theta^-_{\bar{x}\|\bar{u}}$

In general, if the parents of testing node X have n states, the selection process may yield 2^n distinct CPTs.

2.1. Syntax

A TBN is a directed acyclic graph (DAG) with two types of nodes: *regular* and *testing,* each having a conditional probability table (CPT). Root nodes are always regular. Consider a node X with parents $\mathbf{U}$.

- If X is a regular node, its CPT is said to be *regular* and has a parameter $\theta_{x|\mathbf{u}} \in [0,1]$ for each state x of node X and state $\mathbf{u}$ of its parents $\mathbf{U}$, where $\sum_x \theta_{x|\mathbf{u}} = 1$ (these are the CPTs used in BNs).

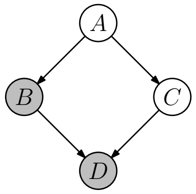


Figure 1. A TBN with binary nodes. Testing nodes are shaded. In a BN, we need 18 parameters to specify the network: 2 for A, 4 for each of B, C and 8 for D. For the TBN, we need 30 parameters: 4 additional parameters for B and 8 additional parameters for D. We also need 2 thresholds for B and 4 thresholds for D.

- If X is a testing node, its CPT is said to be *testing* and has a threshold $T_{X|\mathbf{u}} \in [0,1]$ for each state $\mathbf{u}$ of parents $\mathbf{U}$. It also has two parameters $\theta^+_{x|\mathbf{u}} \in [0,1]$ and $\theta^-_{x|\mathbf{u}} \in [0,1]$ for each state x of node X and state $\mathbf{u}$ of its parents $\mathbf{U}$, where $\sum_x \theta^+_{x|\mathbf{u}} = 1$ and $\sum_x \theta^-_{x|\mathbf{u}} = 1$.

The parameters of a regular CPT are said to be *static* and the ones for a testing CPT are said to be *dynamic.*

Consider a node that has m states and its parents have n states. If the node is regular, its CPT will have $m \cdot n$ static parameters. If it is a testing node, its CPT will have n thresholds and $2 \cdot m \cdot n$ dynamic parameters; see Figure 1. As we shall discuss later, the thresholds and parameters of a TBN can be learned discriminatively from labeled data.

2.2. Semantics

A testing CPT corresponds to a set of regular CPTs, one of which is selected based on the given evidence. Once a regular CPT is selected from each testing CPT, the TBN transforms into a BN. Hence, a TBN over DAG G represents a set of BNs over DAG G, one of which is selected based on the given evidence. It is this selection process that determines the semantics of TBNs. We define this process next based on soft evidence, which includes hard evidence.

Soft evidence on node X with states $x_1, \ldots, x_k$ is specified using *likelihood ratios* $\lambda_1, \ldots, \lambda_k$ (Pearl, 1988). Without loss of generality, we require $\lambda_1 + \ldots + \lambda_k = 1$ so $\lambda_i = 1$ corresponds to hard evidence $X = x_i$. When node X is binary, soft evidence reduces to a single number $\lambda_x \in [0,1]$ since $\lambda_{\bar{x}} = 1 - \lambda_x$. We use Λ to denote all soft evidence.

We next show how to select a BN from a TBN using evidence Λ, thereby defining the semantics of TBNs.

Definition 1 *Consider DAG G and a topological ordering $X_1, \ldots, X_n$ of its nodes, which places non-testing nodes before testing nodes. Define DAGs $G_1, \ldots, G_{n+1}$ such that G_1 is empty and G_{i+1} is obtained by adding node X_i to G_i and connecting it to its parents (hence, $G_{n+1} = G$).*

[2] (Choi & Darwiche, 2018) used evidence on X's ancestors, but we will generalize this later to include more evidence.

[3] Testing can take other forms such as $> T$, $\leq T$ or $< T$.

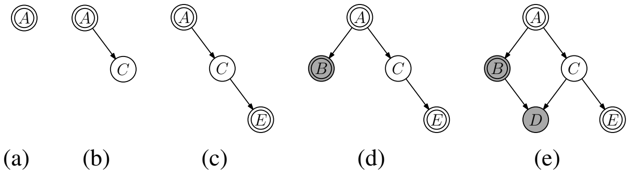


Figure 2. Testing nodes are shaded (B and D) and evidence nodes are double-circled (A and E).

Figure 2 depicts an example of this DAG sequence, using the topological ordering A, C, E, B, D.

Definition 2 *Given TBN G, evidence Λ and DAGs $G_1, \ldots, G_{n+1}$, the selected BN G_{n+1} has the following CPTs. If node X_i is regular, its CPT is copied from the TBN, otherwise it is selected based on the posterior $P_i(\mathbf{U}_i|\Lambda_i)$. Here, $P_i(.)$ is the distribution of BN G_i, $\mathbf{U}_i$ are the parents of X_i and Λ_i is evidence on the ancestors of X_i.*

This definition follows (Choi & Darwiche, 2018) by using ancestral evidence when selecting CPTs, but we will generalize this later. CPTs are selected as discussed earlier:[4]

$$\theta_{x|\mathbf{u}} = \begin{cases} \theta^{+}_{x|\mathbf{u}} & \text{if } P_i(\mathbf{u}_i|\Lambda_i) \geq T_{X|\mathbf{u}} \\ \theta^{-}_{x|\mathbf{u}} & \text{otherwise.} \end{cases}$$

The selected BN is invariant to the specific total order used in Definition 1. Moreover, it has the same structure as the TBN and can be used to answer any query as long as it based on the same evidence Λ used to select the BN. If the evidence changes, a new BN needs to be selected.

2.3. Testing Arithmetic Circuits

A BN query can be computed using an *Arithmetic Circuit (AC),* which is compiled from a BN (Darwiche, 2003; Choi & Darwiche, 2017). A TBN query can be computed using a *Testing Arithmetic Circuit (TAC),* which is compiled from a TBN (Choi & Darwiche, 2018; Choi et al., 2018).

A TAC is an AC that includes *testing units.* A testing unit has two inputs, x and T, and two parameters, θ^+ and θ^-. Its output is computed as follows:[5]

$$f(x, T) = \begin{cases} \theta^{+} & \text{if } x \geq T \\ \theta^{-} & \text{otherwise.} \end{cases}$$

Figure 3(b) depicts a TAC that computes a query on the TBN in Figure 3(a). The TAC inputs $(\lambda_a, \lambda_{\bar{a}})$ and $(\lambda_c, \lambda_{\bar{c}})$ represent soft evidence Λ on nodes A and C. Its outputs represent the marginal $P(B, \Lambda)$. All other TAC inputs correspond to TBN parameters and thresholds: 2 static parameters for node A, 4 static parameters for node C, in addition to 8 dynamic parameters and 2 thresholds for node B. The parameters and thresholds of a TAC can be learned from labeled data using gradient descent (Choi et al., 2018).

2.4. Expressiveness

TBN queries are universal approximators, which means that any continuous function $f(x_1, \ldots, x_n)$ from $[0,1]^n$ to $[0,1]$ can be approximated to an arbitrary error by a TBN query (Choi & Darwiche, 2018; Choi et al., 2018).

The TBN and query used in this result are specific to the given function and error. In practice though, the TBN and query are mandated by modeling and task considerations, so the resulting TBN query may not be as expressive. In general, a TBN joint marginal query computes a piecewise multi-linear function of the evidence (Choi et al., 2018). In particular, the evidential input Λ can be partitioned into regions where the query computes a multi-linear function (like a BN) in each region. Moreover, the number of such regions is linked to the query expressiveness, i.e., its ability to approximate functions from the evidence into a probability.

For reference, neural networks with ReLU activation functions are universal approximators and compute piecewise linear functions (Pascanu et al., 2014; Montúfar et al., 2014). Moreover, there has been work on bounding the number of regions for such functions, depending on the size and depth of neural networks, e.g., (Pascanu et al., 2014; Montúfar et al., 2014; Raghu et al., 2017; Serra et al., 2018).

3. Generalized CPT Selection

TBNs get their expressiveness from the ability to select CPTs based on available evidence, which allows them to compute probabilities based on multiple distributions.

The dependence of CPT selection on only ancestral evidence can be limiting though. For example, in Figure 2, evidence on node E will not participate in selecting the CPT of node B, which reduces expressiveness. In the extreme case of no evidence above a testing node, its CPT selection will not be impacted by the given evidence.

The dependence on ancestral evidence can be relaxed but up to a point as we have the following constraint:

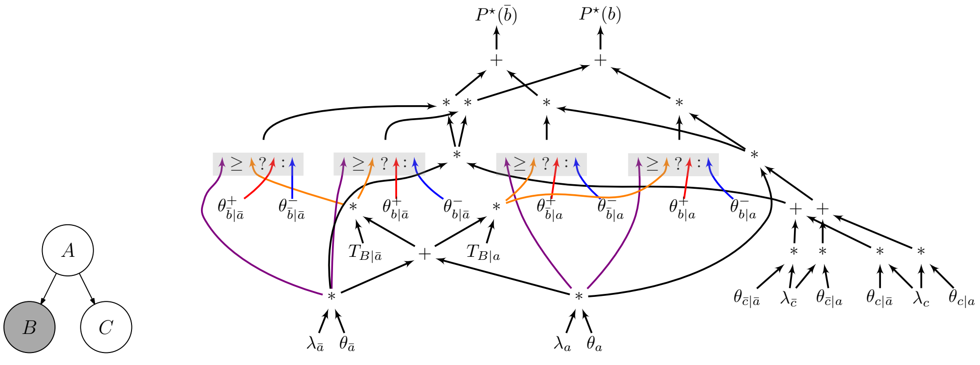
(1) Evidence at/below testing node X_i cannot participate in CPT selection until the CPT of X_i has been selected.

The reason for this constraint is that we need the CPT of node X_i in order to factor evidence at or below it.

We can include some non-ancestral evidence without violat-

[4] The selection can be based on other tests such as $P_i(\mathbf{u}_i|\Lambda_i) > T_{X|\mathbf{u}}$, $P_i(\mathbf{u}_i|\Lambda_i) \leq T_{X|\mathbf{u}}$ or $P_i(\mathbf{u}_i|\Lambda_i) < T_{X|\mathbf{u}}$.

[5] The unit may employ other tests, $x > T$, $x \leq T$ or $x < T$.



(a) TBN (b) TAC

Figure 3. Nodes A, B and C are binary and node B is testing. Nodes $\boxed{x \geq T ? \theta^+ : \theta^-}$ represent testing units.

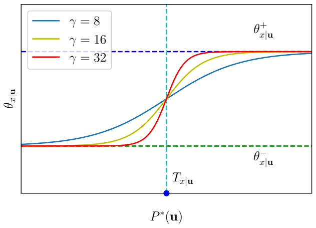


Figure 4. CPT selection using a sigmoid function. The selected parameter $\theta_{x|\mathbf{u}}$ is a weighted average of parameters $\theta^+_{x|\mathbf{u}}$ and $\theta^-_{x|\mathbf{u}}$ ($T_{X|\mathbf{u}}$ is the sigmoid center and γ controls the sigmoid slope).

ing this constraint. In particular, when selecting the CPT of node X_i in Definition 2, we can define Λ_i so it *only excludes* evidence at or below testing nodes X_j that are not ancestors of X_i (the CPTs of nodes X_j are not guaranteed to have been selected at that point). Using this method in Figure 2, evidence on E will now participate in selecting the CPT of node B. However, this evidence cannot participate in this selection if node C was also testing.

The selected BN according to this method is also invariant to the specific total order used in Definition 1.

Before we close this section, we note that the selection of CPTs based on threshold test, e.g., $P(.|.) \geq T$, is not strictly needed. Threshold tests are both simple and sufficient for universal approximation. However, one can employ more general and refined selection schemes, which can also facilitate the learning of TAC parameters and thresholds using gradient descent methods. The main requirement is that the selection process uses only the posterior on parents to make its decisions. For example, one can use a sigmoid function to select CPTs as shown in Figure 4 and detailed in (Choi & Darwiche, 2018; Choi et al., 2018). This leads to TACs with sigmoid units instead of testing units.

4. Conditional Independence

We will now discuss conditional independence in TBNs and whether it can be inferred from d-separation as in BNs.

Our focus is on *hard evidence* using the following key notation. Given a TBN and evidence $\mathbf{e}$, we use $P^{\mathbf{e}}(.)$ to denote the distribution of the selected BN under evidence $\mathbf{e}$. We also use $Q(q||\mathbf{e})$ to denote a TBN query that computes the probability of q given evidence $\mathbf{e}$. Evaluating TBN query $Q(q||\mathbf{e})$ is a two step process: we first select the distribution $P^{\mathbf{e}}(.)$ and then use it to compute the probability $P^{\mathbf{e}}(q|\mathbf{e})$.

We now define conditional independence in TBNs. In what follows, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are disjoint variable sets and $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ are their corresponding instantiations.

Definition 3 *For a TBN, we say* $\mathbf{X}$ *is independent of* $\mathbf{Y}$ *given* $\mathbf{Z}$ *iff* $Q(\mathbf{x}||\mathbf{z}) = Q(\mathbf{x}||\mathbf{zy})$ *for all* $\mathbf{x}$, $\mathbf{y}$ *and* $\mathbf{z}$.

That is, independence holds when $P^{\mathbf{z}}(\mathbf{x}|\mathbf{z}) = P^{\mathbf{zy}}(\mathbf{x}|\mathbf{zy})$. The selected distributions $P^{\mathbf{z}}$ and $P^{\mathbf{zy}}$ may be distinct, but must still assign the same probability to $\mathbf{x}|\mathbf{z}$ and $\mathbf{x}|\mathbf{zy}$, respectively. In BN independence, the two sides of the equality assume the same distribution, which is induced by the same set, yet any set, of CPTs. In TBN independence, the distributions $P^{\mathbf{z}}$ and $P^{\mathbf{zy}}$ may be induced by different CPTs.

In BNs, evidence may change probabilities. In TBNs, evidence may also change the selected CPTs.

Definition 4 *For a TBN, we say the selected CPTs for nodes* $\mathbf{X}$ *are independent of* $\mathbf{Y}$ *given* $\mathbf{Z}$ *iff they are the same under evidence* $\mathbf{z}$ *or evidence* $\mathbf{zy}$*, for all* $\mathbf{y}$ *and* $\mathbf{z}$.

We are interested in the relationship between d-separation and TBN independence, for both selected CPTs and probabilities. In fact, to prove that certain probabilities will not change in a TBN due to evidence, we will have to prove that the selection of all relevant CPTs will not change either.

4.1. d-separation in BNs

We first prove some results about d-separation in BNs, which are instrumental for reasoning about d-separation in TBNs.

Definition 5 *A proper subset of a DAG G is obtained by successively removing some leaf nodes from G.*

A proper subset can also be obtained by removing some nodes and all their descendants from G. In Definition 1, each DAG G_i is a proper subset of DAG G_j for $j > i$.

The following proposition shows how d-separation in DAG G can be used to infer d-separation in its proper subsets.

Proposition 1 *If $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ and $G^\star$ is a proper subset of G, then $dsep_{G^\star}(\mathbf{X}^\star, \mathbf{Z}^\star, \mathbf{Y}^\star)$, where $\mathbf{X}^\star$, $\mathbf{Y}^\star$, $\mathbf{Z}^\star$ are the subsets of $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ in DAG $G^\star$.*

The following proposition identifies evidence that does not impact the parents posterior of a node, which is essential for showing it does not impact the selected CPT of that node.

Proposition 2 *If $dsep_G(X, \mathbf{Z}, \mathbf{Y})$, then $dsep_G(\mathbf{U} \setminus \mathbf{Z}, \mathbf{Z}, \mathbf{Y})$, where $\mathbf{U}$ are the parents of node X in G.*

The following proposition identifies CPTs that are irrelevant to a particular query. If a CPT is irrelevant to a query, then the query is not impacted by how the CPT is selected.

Proposition 3 *If $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ and $P(\mathbf{x}|\mathbf{zy})$ depends on the CPT of node T, then $dsep_G(T, \mathbf{Z}, \mathbf{Y})$.*

Consider a BN $Y \rightarrow T_1 \rightarrow Z \rightarrow T_2 \rightarrow X$. Since $dsep(X, Z, Y)$ and not $dsep(T_1, Z, Y)$, the CPT of node T_1 is irrelevant to query $P(x|zy)$. Hence, changing the CPT of node T_1 will not impact the query $P(x|zy)$ (or $P(x|z)$ since $P(x|zy) = P(x|z)$).

4.2. d-separation in TBNs

We next show that d-separation implies conditional independence in TBNs. Our result is based on CPT selection as given by Definition 2, except that the evidence Λ_i used to select a CPT for node X_i is not restricted to being ancestral. In particular, all we assume is that evidence Λ_i is the projection of evidence Λ on some proper subset of the BN G_i used to select the CPT of node X_i. The methods we discussed for evidence inclusion satisfy this condition. Moreover, a method that satisfies this condition cannot violate Constraint (1) from Section 3.

We start by the impact of d-separation on CPT selection.

Theorem 1 *If $dsep_G(X, \mathbf{Z}, \mathbf{Y})$ in TBN G, then the selected CPT of node X is independent of $\mathbf{Y}$ given $\mathbf{Z}$.*

We are now ready for our main theorem: one can infer conditional independence from d-separation in TBNs.

Theorem 2 *If $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ in TBN G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$.*

Theorem 2 implies that the Markovian assumption is satisfied by TBNs: Every node is independent of its non-descendants given its parents. It also implies that Markov blankets apply in TBNs: Given the parents, children and spouses of a node, it becomes independent of all other nodes.

5. Learning with Incomplete Models

We will show in this section how the expressiveness of TBNs can be used to alleviate a common and practical problem: Learning with incomplete models. We will focus on the task of discriminative learning. That is, our data contains labeled examples of the form $< \Lambda, p >$, where Λ is a soft evidence vector and p is the corresponding probability. Our goal is to learn the function f that generated this labeled data (function f maps evidence to a probability).

Our assumption is that the function f corresponds to a query on a BN G. However, we are unaware of some of the nodes or edges in this data-generating model G, so we are using an incomplete BN structure $G^\star$ to learn function f.

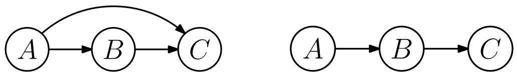
Normally, this task can be accomplished by compiling the structure of BN $G^\star$ into an AC that computes the query of interest (Darwiche, 2003; Choi & Darwiche, 2017). That is, the AC takes the evidence vector Λ as input and generates the sought probability as an output. The AC parameters correspond to parameters in the BN $G^\star$ and can be trained using gradient descent (not all parameters of the BN $G^\star$ may be relevant to the query of interest).

Since BN $G^\star$ misses some nodes or edges from the data-generating BN G, we will next show that the AC compiled from $G^\star$ may not be able to represent the data-generating function f (for any choice of parameters). Moreover, we will show that a TAC compiled from a TBN $G^\star$ is provably a better approximator of the data-generating function f.

5.1. The Functional Form of Marginal Queries

Our first step is to look into the form of function f. For simplicity, we will assume binary variables so soft evidence on a node is captured by a single number $\lambda \in [0, 1]$.

We will distinguish between a *function*, a *functional form* and a *constrained functional form (CFF)*. For example, $f(\lambda) = \lambda - 1$ is a function *admitted* by functional form $f(\lambda) = A\lambda + B$ ($A = 1$, $B = -1$). A functional form is *constrained* iff its constants must satisfy some constraints. For example, if $A = \gamma^2 - (1-\gamma)^2$ and $B = (1-\gamma)^2$ for some $\gamma \in [0, 1]$, then the functional form $f(\lambda) = A\lambda + B$ is constrained. This CFF admits the function $f(\lambda) = 1 - \lambda$ ($\gamma = 0$) but not $f(\lambda) = \lambda - 1$ (B cannot be negative).



(a) true model (b) incomplete model

Figure 5. Missing edge.

Let $f_1(\lambda_1, \ldots, \lambda_k)$ and $f_2(\lambda_1, \ldots, \lambda_k)$ be two CFFs. We say that $f_2(\lambda_1, \ldots, \lambda_k)$ is *less expressive* than $f_1(\lambda_1, \ldots, \lambda_k)$ iff the set of functions admitted by f_2 is a strict subset of the set of functions admitted by f_1.

Marginal BN queries induce constrained functional forms. In particular, for soft evidence $\lambda_1, \ldots, \lambda_k$, a joint marginal query induces a constrained multi-linear function $f(\lambda_1, \ldots, \lambda_k)$ and a conditional marginal query induces a constrained quotient of two multi-linear functions $g(\lambda_1, \ldots, \lambda_k)$. The constraints depend on the BN topology and location of evidence and query variables.

5.2. Missing Nodes and Edges

Consider Figure 5, where A and B are evidence nodes and C is a query node. Model M_2 in Figure 5(b) results from missing edge $A \rightarrow C$ in the true model M_1 of Figure 5(a). Assuming all variables are binary, we have:

$$\begin{aligned}
P_1(c, \Lambda) &= [\theta_a \theta_{b|a} \theta_{c|ab}]\lambda_a \lambda_b + [\theta_a \theta_{\bar{b}|a} \theta_{c|a\bar{b}}]\lambda_a \lambda_{\bar{b}} + \\
&\quad [\theta_{\bar{a}} \theta_{b|\bar{a}} \theta_{c|\bar{a}b}]\lambda_{\bar{a}} \lambda_b + [\theta_{\bar{a}} \theta_{\bar{b}|\bar{a}} \theta_{c|\bar{a}\bar{b}}]\lambda_{\bar{a}} \lambda_{\bar{b}} \\
P_1(\bar{c}, \Lambda) &= [\theta_a \theta_{b|a} \theta_{\bar{c}|ab}]\lambda_a \lambda_b + [\theta_a \theta_{\bar{b}|a} \theta_{\bar{c}|a\bar{b}}]\lambda_a \lambda_{\bar{b}} + \\
&\quad [\theta_{\bar{a}} \theta_{b|\bar{a}} \theta_{\bar{c}|\bar{a}b}]\lambda_{\bar{a}} \lambda_b + [\theta_{\bar{a}} \theta_{\bar{b}|\bar{a}} \theta_{\bar{c}|\bar{a}\bar{b}}]\lambda_{\bar{a}} \lambda_{\bar{b}}
\end{aligned}$$

Noting that $\lambda_{\bar{a}} = 1 - \lambda_a$ and $\lambda_{\bar{b}} = 1 - \lambda_b$, and setting

β_1	β_2	β_3	β_4	β_5	β_6	β_7
θ_a	$\theta_{b\|a}$	$\theta_{b\|\bar{a}}$	$\theta_{c\|ab}$	$\theta_{c\|a\bar{b}}$	$\theta_{c\|\bar{a}b}$	$\theta_{c\|\bar{a}\bar{b}}$

we get the following CFF for computing the posterior on $C = c$ given soft evidence on nodes A and B:

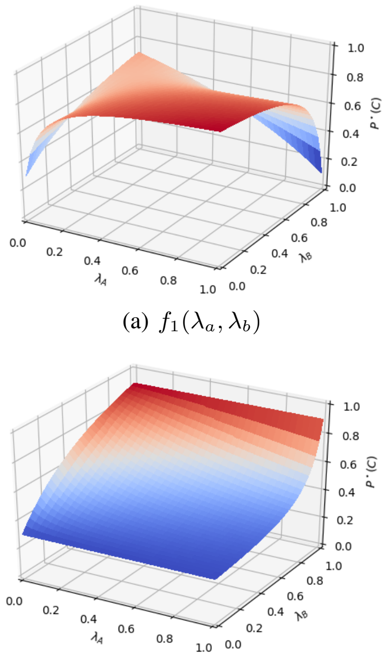
$$f_1(\lambda_a, \lambda_b) = P_1(c|\Lambda) = \frac{\mu_1 \lambda_a \lambda_b + \mu_2 \lambda_a + \mu_3 \lambda_b + \mu_4}{\mu_5 \lambda_a \lambda_b + \mu_6 \lambda_a + \mu_7 \lambda_b + \mu_8}$$

where coefficients $\mu_1, \ldots, \mu_8$ are determined by the *seven* independent parameters $\beta_1, \ldots, \beta_7$ in $[0, 1]$ ($\bar{\beta}_i = 1 - \beta_i$):

$$\begin{array}{lrrrr}
\mu_1 = & \beta_1\beta_2\beta_4 & -\beta_1\bar{\beta}_2\beta_5 & -\bar{\beta}_1\beta_3\beta_6 & +\bar{\beta}_1\bar{\beta}_3\beta_7 \\
\mu_2 = & & \beta_1\bar{\beta}_2\beta_5 & & -\bar{\beta}_1\bar{\beta}_3\beta_7 \\
\mu_3 = & & & \bar{\beta}_1\beta_3\beta_6 & -\bar{\beta}_1\bar{\beta}_3\beta_7 \\
\mu_4 = & & & & \bar{\beta}_1\bar{\beta}_3\beta_7 \\
\mu_5 = & \beta_1\beta_2 & -\beta_1\bar{\beta}_2 & -\bar{\beta}_1\beta_3 & +\bar{\beta}_1\bar{\beta}_3 \\
\mu_6 = & & \beta_1\bar{\beta}_2 & & -\bar{\beta}_1\bar{\beta}_3 \\
\mu_7 = & & & \bar{\beta}_1\beta_3 & -\bar{\beta}_1\bar{\beta}_3 \\
\mu_8 = & & & & \bar{\beta}_1\bar{\beta}_3
\end{array}$$

Similarly, we get a CFF for model M_2 in Figure 5(b):

$$f_2(\lambda_a, \lambda_b) = P_2(c|\Lambda) \quad = \quad \frac{\nu_1 \lambda_a \lambda_b + \nu_2 \lambda_a + \nu_3 \lambda_b + \nu_4}{\nu_5 \lambda_a \lambda_b + \nu_6 \lambda_a + \nu_7 \lambda_b + \nu_8}$$



(a) $f_1(\lambda_a, \lambda_b)$

(b) $f_2(\lambda_a, \lambda_b)$

Figure 6. Functions that compute the probability of $C = c$ given soft evidence on A and B in the models of Figure 5.

We are omitting the constraints on coefficients $\nu_1, \ldots, \nu_8$ for space limitations.

Every function admitted by f_2 satisfies the following: if input λ_b is set to 0 or 1 (i.e., hard evidence), the function output will become independent of input λ_a. Figure 6(b) provides an example, but this can be shown more generally since $f_2|_{\lambda_b=0} = \theta_{c|\bar{b}}$ and $f_2|_{\lambda_b=1} = \theta_{c|b}$. However, there are functions admitted by f_1 that do not satisfy this constraint as shown in Figure 6(a). Hence, CFF f_2 is less expressive than f_1. The two CFFs are equally expressive if $\beta_4 = \beta_6$ and $\beta_5 = \beta_7$. In this case, $\theta_{c|ab} = \theta_{c|\bar{a}b}$ and $\theta_{c|a\bar{b}} = \theta_{c|\bar{a}\bar{b}}$, so the edge $A \rightarrow C$ superfluous.

One can similarly show that missing nodes can also lead to losing the ability to represent the data-generating function.

While a TBN for an incomplete structure may also not be able to represent the data-generating function, it is provably a better approximator than a BN over the same structure. Moreover, all approximations generated by a TBN are guaranteed to respect the conditional independences implied by its structure. Hence, the additional expressiveness remains guarded by the available modeling assumptions. Viewing TACs and ACs as constrained functional forms, we now have the following.

Theorem 3 *Consider a BN and a TBN over the same DAG G and consider a corresponding AC and TAC for some query. The TAC is more expressive than the AC.*

We next discuss a class of functions where TBN queries are a better approximator than BN queries.

5.3. Simpson's Paradox

Simpson's paradox is a phenomenon in which a trend appears in several different groups of data but disappears or reverses when these groups are combined; see, e.g., (Malinas & Bigelow, 2016; Pearl, 2014).

Definition 6 *A distribution $P(A, B, C)$ exhibits Simpson's paradox if $P(c|a) \leq P(c|\bar{a})$ but $P(c|a, b) > P(c|\bar{a}, b)$ and $P(c|a, \bar{b}) > P(c|\bar{a}, \bar{b})$.*

That is, the probability of c given a is no greater than that of c given $\bar{a}$, but this reverses under every value of B. A function $f(\lambda_a, \lambda_b)$ that computes the probability of c given soft evidence on A and B exhibits Simpson's paradox if $f(1, 1/2) \leq f(0, 1/2)$, $f(1, 1) > f(0, 1)$ and $f(1, 0) > f(0, 0)$ since a soft evidence of $1/2$ amounts to no evidence.

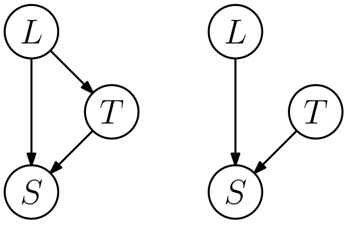
Simpson's paradox typically arises when we have two causes A and B, for some effect C, which are not independent. For example, C could be an admission decision, where A represents gender and B represents the department applied to. Normally, one would expect A and B to be independent, but it is possible that an applicant's gender influences which department they may apply to. When this influence is missed, two things may happen. First, the data may look surprising implying a paradox. For example, the data may show that each department has a higher admission rate for females, but the overall admission rate for males is higher. While this may seem paradoxical, it can be explained away by the fact that females apply to competitive departments with higher rates than males. The second thing that may happen is that a model that misses the direct influence between A and B may not be able to learn Simpson's paradox even though it is exhibited in the data.

Proposition 4 *Consider a BN with edges $A \rightarrow C$ and $B \rightarrow C$, where all variables are binary. Under any parameterization of the BN, if $P(c|a, b) > P(c|\bar{a}, b)$ and $P(c|a, \bar{b}) > P(c|\bar{a}, \bar{b})$, then $P(c|a) > P(c|\bar{a})$.*

Hence, if the edge between A and B is missed, then the BN model will not be able to capture Simpson's paradox if exhibited in the data. As it turns out, however, TBNs can still learn this pattern even if the edge is missed. We next provide a concrete example illustrating this phenomenon.

This is a real-world example comparing the success rates of two treatments for kidney stones (`https://en.wikipedia.org/wiki/Simpson%27s_paradox`).

The following data shows the success rates of treatments:



(a) true (b) incomplete

Figure 7. Kidney stone model: L is whether stone is large (yes, no), T is treatment (A, B) and S is treatment success (yes, no).

	Treatment A	Treatment B
	Group 1	Group 2
Small Stones	**93%** (81/87)	87% (234/270)
	Group 3	Group 4
Large Stones	**73%** (192/263)	69% (55/80)
Both	78% (273/350)	**83%** (289/350)

The paradoxical reading of the above table: Treatment A is more effective when used on small stones and also when used on large stones. Yet, treatment B is more effective when considering both sizes at the same time. Explanation: Doctors favor treatment B for small stones. Hence, treatment B suggests a less severe case (small stone).

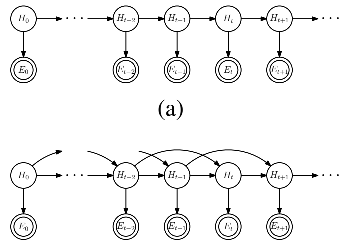
Figure 7(a) depicts a corresponding BN, which parameters can be computed from the above table (maximum-likelihood parameters). Using this data-generating BN, $P(S = yes|T = A) = 78\%$ and $P(S = yes|T = B) = 83\%$.

We compiled an AC and a TAC from the incomplete structure in Figure 7(b) and trained them using nine examples:

		Labeled Data			Predictions	
λ_L	λ_T	Large	Treatment	BN	AC	TAC
1.0	1.0	Yes	A	73.0	71.4	73.0
0.0	1.0	No	A	93.0	91.4	93.0
0.5	1.0	?	A	**77.9**	**81.1**	**78.0**
1.0	0.0	Yes	B	69.0	71.5	69.0
0.0	0.0	No	B	87.0	88.6	87.1
0.5	0.0	?	B	**82.9**	**79.8**	**83.1**
1.0	0.5	Yes	?	72.1	71.5	72.1
0.0	0.5	No	?	88.4	88.7	88.3
0.5	0.5	?	?	80.4	79.8	80.3

The TAC captures Simpson's paradox despite the missing edge $L \rightarrow T$: The overall success rate for treatment B is higher than for treatment A, but this is reversed when considering stone size. The AC fails to capture this pattern (as expected): the success rate for treatment B is lower overall and for small stones, but is higher for large stones.

Overall, the TAC predictions are much better than the AC predictions and are very close to the ground truth. It is interesting that this is achieved even though L and T are independent in the TAC/TBN by Theorem 2.



(a)

(b)

Figure 8. HMM and second-order HMM.

6. A Case Study: Testing HMMs

Figure 8(a) illustrates a Hidden Markov Model (HMM), with hidden nodes H_t and observables E_t. To define an HMM, we need an initial distribution $P(H_0)$, a transition model $P(H_t \mid H_{t-1})$ and an emission model $P(E_t \mid H_t)$.

We want to learn a function that computes the state of hidden node H_n given evidence on $E_0, \dots, E_{n-1}$, where n is the length of the HMM. We assume, however, that labeled data is generated from a higher order HMM in which each hidden node H_t can depend on more than the previous hidden node H_{t-1}. Figure 8(b) depicts a second-order HMM, in which a hidden node H_t has H_{t-2} and H_{t-1} as its parents, $t \geq 2$.

We simulated examples from a third-order HMM and trained both an HMM and a Testing HMM using the structure in Figure 8(a). That is, we pretended that we were unaware of the edges $H_{t-2} \to H_t$ and $H_{t-3} \to H_t$. Training records $< e_0, \dots, e_{n-1} \; : h_n >$ were sampled from the joint distribution of the third-order HMM (data-generating model). The cross entropy loss was used to train both the HMM and the Testing HMM using an AC and a TAC, respectively. Our goal was to demonstrate the extent to which a Testing HMM can compensate the modeling error, i.e., the missing dependencies of H_t on H_{t-2} and H_{t-3}.

We considered all transition models for third-order HMMs such that $P(h_t \mid h_{t-3}, h_{t-2}, h_{t-1})$ is either 0.95 or 0.05. We assumed binary variables and a chain of length 8. We used uniform initial distributions and emission model $P(h_t \mid e_t) = P(\bar{h_t} \mid \bar{e_T}) = 0.99$. There were 256 third-order HMMs satisfying these conditions. We fit an AC (HMM) and a TAC (Testing HMM) using data simulated from each, with sigmoid selection in the TAC. We used data sets with $16,384$ records for each run and 5-fold cross validation to report prediction accuracy as shown in Figure 9. The x-axis measures the accuracy of the HMM, and the y-axis measures the accuracy of the Testing HMM. There are 256 data points in Figure 9, each representing a distinct third-order HMM used. The error bar around each data point represents the standard deviation over the 5-fold cross validation.

In Figure 9, $178/256$ points are above the dashed diagonal line, indicating a better prediction accuracy for the Testing HMM over the HMM. Moreover, 82 of the data points obtain accuracies above 95% for the Testing HMM. This further illustrates the extent to which the Testing HMM can recover from the underlying modeling error.

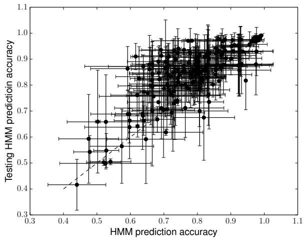


Figure 9. Accuracy of fitting a third-order HMM by an HMM and a Testing HMM.

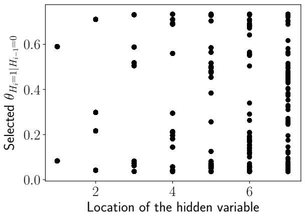


Figure 10. Selected parameters by hidden node H_t in a Testing HMM across all evidence.

We can view a Testing HMM as a set of *heterogeneous* HMMs since a hidden node H_t may select a *different* transition model depending on evidence $e_0, \dots, e_{t-1}$. In contrast, the learned HMM uses the same transition model across all hidden nodes. In Figure 10, we visualize the distinct parameters selected by nodes H_t in the Testing HMMs (across all possible evidence). When t is small, we see fewer distinct parameters as nodes H_t use a limited number of evidence nodes in the test. For larger t, we see that hidden nodes select from a larger set of parameters. This intuitively explains why Testing HMMs are more expressive than HMMs.

7. Conclusion

TBNs were introduced recently, motivated by an expressiveness gap between Bayesian and neural networks. A TBN represents a set of distributions, one of which is selected based on the given evidence and used for reasoning. This makes TBNs more expressive than BNs and as expressive as neural networks. We showed that TBN independence can be inferred from d-separation as in BNs. We also improved the expressiveness of TBN queries by making TBN selection more sensitive to evidence. We finally showed that TBN expressiveness and independence can help alleviate a common and practical problem, which arises when learning from labeled data using incomplete models (i.e., ones that are missing nodes or edges from the data-generating model).

Acknowledgements

This work has been partially supported by NSF grant #IIS-1514253, ONR grant #N00014-18-1-2561 and DARPA XAI grant #N66001-17-2-4032.

References

Choi, A. and Darwiche, A. On relaxing determinism in arithmetic circuits. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning (ICML)*, pp. 825–833, 2017.

Choi, A. and Darwiche, A. On the relative expressiveness of Bayesian and neural networks. In *Proceedings of the 9th International Conference on Probabilistic Graphical Models (PGM)*, pp. 157–168, 2018.

Choi, A., Wang, R., and Darwiche, A. On the relative expressiveness of Bayesian and neural networks, 2018. http://arxiv.org/abs/1812.08957.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

Darwiche, A. A differential approach to inference in Bayesian networks. *Journal of the ACM (JACM)*, 50 (3):280–305, 2003.

Hornik, K., Stinchcombe, M. B., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

Malinas, G. and Bigelow, J. Simpson's paradox. In Zalta, E. N. (ed.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2016.

Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pp. 2924–2932, 2014.

Pascanu, R., Montúfar, G., and Bengio, Y. On the number of inference regions of deep feed forward networks with piece-wise linear activations. In *2nd International Conference on Learning Representations ICLR*, 2014.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. MK, 1988.

Pearl, J. Comment: Understanding Simpson's paradox. *The American Statistician*, 68(1):8–13, 2014.

Raghu, M., Poole, B., Kleinberg, J. M., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning ICML*, pp. 2847–2854, 2017.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. Bounding and counting linear regions of deep neural networks. In *Proceedings of the 35th International Conference on Machine Learning ICML*, pp. 4565–4573, 2018.

A. Proofs

Proposition 1 Suppose $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ holds but $dsep_{G^\star}(\mathbf{X}^\star, \mathbf{Z}^\star, \mathbf{Y}^\star)$ does not hold. There must exist a path in $G^\star$ that connects $\mathbf{X}^\star$ and $\mathbf{Y}^\star$ but is not blocked by $\mathbf{Z}^\star$. This path also connects $\mathbf{X}$ and $\mathbf{Y}$ in G. Moreover, it is not blocked by $\mathbf{Z}$ since no variable in $\mathbf{Z} \setminus \mathbf{Z}^\star$ can be on this path. This is a contradiction with $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. Hence $dsep_{G^\star}(\mathbf{X}^\star, \mathbf{Z}^\star, \mathbf{Y}^\star)$ holds. □

Proposition 2 Suppose $dsep_G(X, \mathbf{Z}, \mathbf{Y})$ holds but $dsep_G(\mathbf{U} \setminus \mathbf{Z}, \mathbf{Z}, \mathbf{Y})$ does not hold. We have $\mathbf{U} \cap \mathbf{Y} = \emptyset$ by $dsep_G(X, \mathbf{Z}, \mathbf{Y})$. Moreover, there must exist a path connecting some $U \in \mathbf{U} \setminus \mathbf{Z}$ and $\mathbf{Y}$ that is not blocked by $\mathbf{Z}$. If X is on this path, then we have a path connecting X and $\mathbf{Y}$ that is not blocked by $\mathbf{Z}$. Otherwise, augmenting this path with the edge $X \leftarrow U$ leads to a path with the same properties. Either case contradicts $dsep_G(X, \mathbf{Z}, \mathbf{Y})$. □

Proposition 3 Suppose $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. A leaf node outside $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ is irrelevant to $P(\mathbf{x}|\mathbf{zy})$, so repeatedly remove all such leaf nodes. Prune edges outgoing from nodes $\mathbf{Z}$ as this does not change the value of $P(\mathbf{x}|\mathbf{zy})$ either. Nodes $\mathbf{X}$ are now disconnected from $\mathbf{Y}$. If $dsep_G(T, \mathbf{Z}, \mathbf{Y})$ does not hold, then node T is connected to $\mathbf{Y}$ and disconnected from $\mathbf{X}$ so $P(\mathbf{x}|\mathbf{zy})$ cannot depend on the CPT of T. Hence, if $P(\mathbf{x}|\mathbf{zy})$ depends on the CPT of T, then $dsep_G(T, \mathbf{Z}, \mathbf{Y})$. □

The next proof uses the following corollary of Proposition 1.

Corollary 1 *If $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$, G is a proper superset of G_i and G_i is a proper superset of G_j, then $dsep_{G_i}(\mathbf{X}_j, \mathbf{Z}_j, \mathbf{Y}_j)$, where $\mathbf{X}_j$, $\mathbf{Y}_j$, $\mathbf{Z}_j$ are the subsets of $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ in DAG G_j.*

Theorem 1 The proof is by induction on DAGs $G_1, \ldots, G_{n+1}$ in Definition 1. We will show: if $dsep_{G_i}(X_k, \mathbf{Z}_i, \mathbf{Y}_i)$, the CPT for node X_k is independent of $\mathbf{Y}_i$ given $\mathbf{Z}_i$. Here, $\mathbf{Y}_i$ and $\mathbf{Z}_i$ are projections of $\mathbf{Y}$ and $\mathbf{Z}$ on some proper subset of G_i. The theorem statement is for $i = n + 1$.

This holds trivially for G_1 (empty). Consider G_i for $i > 1$ and assume this holds for G_j, where $j < i$. Suppose $dsep_{G_i}(X_k, \mathbf{Z}_i, \mathbf{Y}_i)$ for $k < i$. Then G_k is a proper subset of G_i and contains parents $\mathbf{U}_k$ of node X_k. We have $dsep_{G_i}(\mathbf{U}_k \setminus \mathbf{Z}_i, \mathbf{Z}_i, \mathbf{Y}_i)$ by Proposition 2 and $dsep_{G_k}(\mathbf{U}_k \setminus \mathbf{Z}_k, \mathbf{Z}_k, \mathbf{Y}_k)$ by Corollary **??**. The CPT of node X_k is selected in G_k based on $P_k(\mathbf{U}_k|\mathbf{z}_k\mathbf{y}_k)$. Consider node X_m in G_k ($m < k$). By Proposition 3, if the CPT of node X_m is relevant to $P_k(\mathbf{U}_k|\mathbf{z}_k\mathbf{y}_k)$, then $dsep_{G_k}(X_m, \mathbf{Z}_k, \mathbf{Y}_k)$. By the induction hypothesis, this CPT is independent of $\mathbf{Y}_k$ given $\mathbf{Z}_k$. That is, every CPT in G_k that is relevant to $P_k(\mathbf{U}_k|\mathbf{z}_k\mathbf{y}_k)$ is independent of $\mathbf{Y}_k$ given $\mathbf{Z}_k$. Moreover, $P_k(\mathbf{U}_k|\mathbf{z}_k\mathbf{y}_k) = P_k(\mathbf{U}_k|\mathbf{z}_k)$ since $dsep_{G_k}(\mathbf{U}_k \setminus \mathbf{Z}_k, \mathbf{Z}_k, \mathbf{Y}_k)$. Hence, the CPT of node X_k is independent of $\mathbf{Y}_k$ given $\mathbf{Z}_k$ and of $\mathbf{Y}_i$ given $\mathbf{Z}_i$. □

Theorem 2 Assume $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. We show $Q(\mathbf{x}||\mathbf{yz}) = Q(\mathbf{x}||\mathbf{z})$, which reduces to $P^{\mathbf{zy}}(\mathbf{x}|\mathbf{zy}) = P^{\mathbf{z}}(\mathbf{x}|\mathbf{z})$. By Proposition 3, if $P^{\mathbf{zy}}(\mathbf{x}|\mathbf{zy})$ depends on the CPT of some node T, then $dsep_G(T, \mathbf{Z}, \mathbf{Y})$. By Theorem 1, this CPT is independent of $\mathbf{Y}$ given $\mathbf{Z}$. Hence, $P^{\mathbf{zy}}(\mathbf{x}|\mathbf{zy}) = P^{\mathbf{z}}(\mathbf{x}|\mathbf{zy})$. Moreover, $P^{\mathbf{z}}(\mathbf{x}|\mathbf{zy}) = P^{\mathbf{z}}(\mathbf{x}|\mathbf{z})$ since $dsep_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. Therefore, $P^{\mathbf{zy}}(\mathbf{x}|\mathbf{zy}) = P^{\mathbf{z}}(\mathbf{x}|\mathbf{z})$. □

Theorem 3 The TAC can simulate the AC by setting its parameters $\theta^+_{x|\mathbf{u}}$ and $\theta^-_{x|\mathbf{u}}$ to the corresponding parameter $\theta_{x|\mathbf{u}}$ in the AC. Hence, the TAC is no less expressive than the AC. Proposition 4 identifies a class of functions that cannot be represented by an AC. A function in this class is given in Section 5.3 (kidney stones), which can be represented by a corresponding TAC. Hence, the TAC is more expressive than the AC. □

Proposition 4 $P(c|a, b) > P(c|\bar{a}, b)$ and $P(c|a, \bar{b}) > P(c|\bar{a}, \bar{b})$ imply $\theta_{c|ab} > \theta_{c|\bar{a}b}$ and $\theta_{c|a\bar{b}} > \theta_{c|\bar{a}\bar{b}}$. Moreover, $P(c|a) = \theta_b\theta_{c|ab} + \theta_{\bar{b}}\theta_{c|a\bar{b}}$ and $P(c|\bar{a}) = \theta_b\theta_{c|\bar{a}b} + \theta_{\bar{b}}\theta_{c|\bar{a}\bar{b}}$. Hence, $P(c|a) - P(c|\bar{a}) = \theta_b(\theta_{c|ab} - \theta_{c|\bar{a}b}) + \theta_{\bar{b}}(\theta_{c|a\bar{b}} - \theta_{c|\bar{a}\bar{b}}) > 0$. □