

Acquisition of Causal Models for Local Distributions in Bayesian Networks

Yang Xiang and Minh Truong, University of Guelph, Canada

Abstract—To specify a Bayesian network, a local distribution in the form of a conditional probability table, often of an effect conditioned on its n causes, needs to be acquired, one for each non-root node. Since the number of parameters to be assessed is generally exponential in n , improving the efficiency is an important concern in knowledge engineering. Non-impeding noisy-AND (NIN-AND) tree causal models reduce the number of parameters to being linear in n , while explicitly expressing both reinforcing and undermining interactions among causes. The key challenge in NIN-AND tree modeling is the acquisition of the NIN-AND tree structure. In this work, we formulate a concise structure representation and an expressive causal interaction function of NIN-AND trees. Building on these representations, we propose two structural acquisition methods, which are applicable to both elicitation-based and machine learning-based acquisitions. Their accuracy is demonstrated through experimental evaluations.



1 INTRODUCTION

To specify a Bayesian network (BN), for each non-root node, a local distribution in the form of a conditional probability table (CPT) needs to be acquired. A BN is often constructed in the causal direction, in which case a CPT is defined over an effect conditioned on its n causes. In general, the number of parameters to be assessed for specifying a CPT is exponential in n . The noisy-OR [Pea88] and several extensions, e.g., [HB96], [GD00], [LG04], reduce the number to being linear in n . However, these methods are limited to expressing reinforcing causal interactions [XJ07]. Other methods also exist, e.g., the binary factorization [NCF12].

NIN-AND tree [XJ07] causal models, including their DeMorgan special cases [MD08], extend the noisy-OR by explicitly expressing reinforcing and undermining causal interactions as well as their mixtures at multiple levels. Each model is specified by a linear (in n) number of parameters and a tree of NIN-AND gates whose number is linear in n . A common noisy-AND gate is impeding, where falsity of any cause obstructs the causal influence of remaining causes, forcing the causal probability to zero. An NIN-AND gate is non-impeding and its causal probability is non-zero if at least one cause is true.

In an NIN-AND tree model, the tree topology encodes types and the order of interactions among causes. To acquire local distributions in BNs by NIN-AND tree modeling, the main challenge is structural acquisitions, namely, acquisitions of NIN-AND trees. Elicitation of the topology requires nontrivial training of a domain expert on the syntax and semantics of NIN-AND tree models. It also demands nontrivial mental exercise by the expert in order to articulate the partial order of interactions among causes. The usability of NIN-AND tree causal modeling will be enhanced if such training and mental exercise can be

avoided. This paper presents several advancements towards this goal.

We formulate a concise structure representation of NIN-AND trees, called root-labeled trees (RLTs), and present a new algorithm for their enumeration. We define a function, named PCI, that captures *pairwise causal interactions* encoded by an NIN-AND tree. We prove the expressiveness of PCI functions in the sense that they uniquely identify their generating NIN-AND trees. Based on the RLT enumeration and PCI functions, we propose two methods to acquire NIN-AND trees indirectly through a small subset of acquired probabilities. These numerical parameters can be elicited from a domain expert without above-mentioned training and mental exercise. They can also be learned from data, thereby bypassing the need of a domain expert. Hence, the methods apply to both elicitation-based and machine learning-based model acquisitions.

It is important to differentiate between acquisition for the acyclic directed graph (DAG) of a BN and that for the topology of an NIN-AND tree model. The DAG encodes conditional independence among BN variables, while the NIN-AND tree encodes causal interactions and causal independence among cause variables in a BN family (including a child variable and its parents). For instance, the building block patterns previously studied facilitate the construction of DAGs [NFN00]. However, this work concerns NIN-AND tree acquisitions. Their relation and difference are illustrated in Section 7.4.

Causal modeling with reinforcement and undermining is also studied under the terms synergy and antagonism [CBS08]. Their approach considers only variable based interactions and produces a partially specified BN. The dual NIN-AND gate (see Section 2) is a special case of synergy and the direct NIN-AND gate is a special case of antagonism. The NIN-

AND tree allows reinforcing (undermining) among variables in a group and undermining (reinforcing) between multiple such groups, and allows multiple levels of such mixtures. Thus, it expresses both variable and group based interactions in a specified and precise manner, which goes beyond the referenced work. Assuming the DAG is specified, the methods proposed enable efficient CPT specification at each node, which leads to a completely defined BN.

The RLT enumeration is an important component of our proposed methods for NIN-AND tree acquisition. Most methods for tree enumeration in the literature do not address this issue. Earlier investigations on RLT enumeration followed an approach of indirect enumeration. A set Ω_n of unlabelled trees of n roots (one leaf) are first enumerated and then each is root-labeled to produce the set Ψ_n of RLTs. Two methods have been proposed under the indirect approach. One is based on local insertion [XLZ09]. Since isomorphic trees may be generated, it requires detection and removal of duplicated trees. Another is based on enumerating partitions of integer n [XZL09], which does not suffer from the duplication generation. A limitation of the indirect approach is that it must deal with mirror-subtrees [XZL09] during root-labeling, whose detection becomes fairly complicated for $n > 7$.

In this paper, we instead take a direct approach for RLT enumeration by extending a method [Fel04] for counting evolutionary trees. This method [Fel04] is closely related to our task of RLT enumeration, although it has several limitations. It considers trees of a single root while RLTs have a single leaf. Its tree format differs from standard graphs, where a link may connect to a single node, instead of two (e.g., Fig. 3.5 [Fel04]). It deals with counting, not enumeration. We extend the method into a new algorithm for directly enumerating RLTs of n roots. In comparison with the indirect approach, the direct approach is not as space-efficient. This is because the indirect approach generates Ψ_n from Ω_n while the direct approach generates Ψ_n from Ψ_{n-1} , and $|\Psi_{n-1}|$ is much larger than $|\Omega_n|$. The advantage of the direct approach is that it does not suffer from the complication of mirror-subtrees and is therefore much simpler to implement.

Whether PCI functions uniquely identify minimal NIN-AND trees is fundamental to the soundness of one of the structure acquisition methods presented here. This issue is studied empirically [XLZ09] by exhaustively testing for $n \leq 10$. The theorem on the expressiveness of PCI functions in this paper resolves the issue formally and conclusively for all n .

The experimental methods used in this work extends those in earlier empirical evaluations of causal models, such as noisy-OR [ZD04] and DeMorgan [MD08]. Their evaluations are based on elicitation only while ours are also enhanced by machine learning-based experiments. Their studies compare between the elicited model CPT and the *observed*

relative frequency parameters, while we compare the acquired model with the true model. This difference enables a refined analysis of impacts of factors contributing to the acquisition process, including the sampling errors, the retention-articulation (RA) errors, as well as the acquisition method used. Our experimental design more accurately evaluates the effectiveness of the proposed methods, since it is the true model, not its observed version (which contains the sampling and RA errors), that is the target of acquisition. We compare the earlier experimental procedure with ours in Section 7.3.

We show that if the RA errors can be properly controlled, conducting machine learning-based experiments is equivalent to carrying out elicitation-based experiments. Since human participants are costly, the machine learning-based experiments are preferable.

Preliminary results on the proposed acquisition methods were previously presented [XTZ⁺11]. The remainder of the paper is organized as follows. The background on NIN-AND tree models is covered in Section 2. Section 3 presents the RLT representation and the direct enumeration algorithm. Section 4 defines PCI functions and proves their expressiveness. The structural acquisition methods are presented in Section 5. Section 6 considers parameter acquisition errors and their control. Section 7 reports the experimental study of the proposed methods.

2 NIN-AND TREE CAUSAL MODELS

An *uncertain cause* is a cause that can produce an effect but does not always do so. Flu is a uncertain cause of fever. We focus on binary variables. Denote an effect by e and a set of causes of e by $X = \{c_1, \dots, c_n\}$. Denote $e = \text{true}$ by e^+ and $e = \text{false}$ by e^- . For each c_i , where $1 \leq i \leq n$, denote $c_i = \text{true}$ (active) by c_i^+ and $c_i = \text{false}$ (inactive) by c_i^- .

A *causal success* is an event that a cause c_i caused e to occur when other causes are false. Denote the event by $e^+ \leftarrow c_i^+$ and its probability by $P(e^+ \leftarrow c_i^+)$. A *causal failure* is an event where e is false, when c_i is true and other causes of e are false. It is denoted by $e^- \leftarrow c_i^+$. Denote the causal success that a set $X = \{c_1, \dots, c_n\}$ of causes caused e by $e^+ \leftarrow c_1^+, \dots, c_n^+$ or $e^+ \leftarrow \underline{x}^+$. Denote the set of *all* causes of e by C .

The CPT $P(e|C)$ relates to causal probabilities as follows. If $C = \{c_1, c_2, c_3\}$, then

$$P(e^+|c_1^+, c_2^+, c_3^-) = P(e^+ \leftarrow c_1^+, c_2^+).$$

A causal probability (the right-hand side) is always equivalent to a conditional probability (the left-hand side). The converse may not be true. C is assumed to include a leak variable (if any) for causes that are left implicit, and hence $P(e^+|c_1^-, c_2^-, c_3^-) = 0$.

Causes reinforce each other if they are collectively at least as effective as when only some are acting. For example, consider the effect of curing cancer

with radiotherapy and chemotherapy being its causes. When both are applied, the probability of curing the cancer is improved. Causes undermine each other if they are collectively less effective. Consider a person's well-being as the effect and taking either one of two desirable jobs as the causes. When taking both, the chance of well-being is reduced due to overstress. If $C = \{c_1, c_2\}$, and c_1 and c_2 undermine each other, the following holds.

$$P(e^+ | c_1^-, c_2^-) = 0, P(e^+ | c_1^+, c_2^-) > 0, P(e^+ | c_1^-, c_2^+) > 0,$$

$$P(e^+ | c_1^+, c_2^+) < \min(P(e^+ | c_1^+, c_2^-), P(e^+ | c_1^-, c_2^+)).$$

Def. 1 defines the two types of causal interactions.

Definition 1. Let $R = \{W_1, W_2, \dots\}$ be a partition of a set X of causes, $R' \subset R$, and $Y = \cup_{W_i \in R'} W_i$. Sets of causes in R reinforce each other iff

$$\forall R' P(e^+ \leftarrow \underline{y}^+) \leq P(e^+ \leftarrow \underline{x}^+),$$

and they undermine each other iff

$$\forall R' P(e^+ \leftarrow \underline{y}^+) > P(e^+ \leftarrow \underline{x}^+).$$

Reinforcing and undermining occur between individual variables as well as sets of them. In individual cases, W_i is a singleton. Otherwise, W_i is a generic set. Consider $X = \{c_1, c_2, c_3, c_4, c_5\}$,

$$W_1 = \{c_1, c_2\}, W_2 = \{c_3, c_4, c_5\}, R = \{W_1, W_2\}.$$

While c_1 and c_2 undermine each other, and so do c_3 , c_4 and c_5 , W_1 and W_2 may reinforce each other.

W_1, \dots, W_m (disjoint subsets of X) satisfy *failure conjunction* iff

$$(e^- \leftarrow \underline{w}_1^+, \dots, \underline{w}_m^+) = (e^- \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^- \leftarrow \underline{w}_m^+).$$

That is, the collective failure is attributed to individual failures. They also satisfy *failure independence* iff

$$\begin{aligned} P((e^- \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^- \leftarrow \underline{w}_m^+)) \\ = P(e^- \leftarrow \underline{w}_1^+) \times \dots \times P(e^- \leftarrow \underline{w}_m^+). \end{aligned} \quad (1)$$

W_1, \dots, W_m (disjoint) satisfy *success conjunction* iff

$$(e^+ \leftarrow \underline{w}_1^+, \dots, \underline{w}_m^+) = (e^+ \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \leftarrow \underline{w}_m^+).$$

That is, the collective success requires individual effectiveness. They also satisfy *success independence* iff

$$\begin{aligned} P((e^+ \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \leftarrow \underline{w}_m^+)) \\ = P(e^+ \leftarrow \underline{w}_1^+) \times \dots \times P(e^+ \leftarrow \underline{w}_m^+). \end{aligned} \quad (2)$$

Causes are undermining when success conjunction and independence hold. Hence, undermining can be modeled by a *direct NIN-AND gate* (Fig. 1 (left)). Each root (top) is a single-causal success and the leaf (bottom) is the causal success in question. Success conjunction is encoded by an AND gate and independence is encoded by the disconnection of roots other than through the gate. The leaf event probability is computed by Eqn. (2).

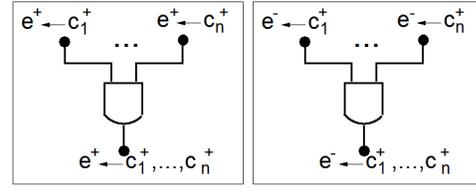


Fig. 1. Direct (left) and dual (right) NIN-AND gates

Causes are reinforcing when failure conjunction and independence hold. Hence, reinforcement can be modeled by a *dual NIN-AND gate* in Fig. 1 (right). Each root is a single-causal failure and the leaf is the causal failure in question. The leaf event probability is computed by Eqn. (1).

By organizing direct and dual gates into a tree, reinforcing, undermining, as well as their mixtures at multiple levels, can be expressed by an NIN-AND tree.

Definition 2. An NIN-AND tree T is a directed tree for an effect e and its active causes $X = \{c_1, \dots, c_n\}$.

An event node (black oval) has an in-degree ≤ 1 and an out-degree ≤ 1 . A gate node has an in-degree ≥ 2 and an out-degree 1. A forward link connects an event and a gate. A negation link has a white oval at the gate end. A direct gate is connected to causal successes only and a dual gate is connected to causal failures only.

Each terminal is an event labelled by $e^+ \leftarrow y^+$ or $e^- \leftarrow y^+$, where $Y \subseteq X$. A single leaf with $y^+ = \underline{x}^+$ connects to the leaf gate. Each root (indexed by i) satisfies $\underline{y}_i^+ \subset \underline{x}^+$, $\underline{y}_j^+ \cap \underline{y}_k^+ = \emptyset$ for $j \neq k$, and $\cup_i \underline{y}_i^+ = \underline{x}^+$.

Example 1. Consider $C = \{c_1, c_2, c_3\}$. Suppose that c_1 and c_3 reinforce each other and collectively they undermine c_2 . The two-level mixture of reinforcing and undermining relative to event $e^+ \leftarrow c_1^+, c_2^+, c_3^+$ is encoded in Fig. 2 (a). The top gate is dual and the leaf gate is direct. The left link into the leaf gate is a negation link.

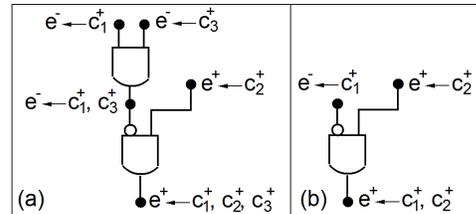


Fig. 2. (a) An NIN-AND tree over $C = \{c_1, c_2, c_3\}$. (b) An NIN-AND tree over $X = \{c_1, c_2\}$.

Example 2. From single-causal probabilities

$$P(e^+ \leftarrow c_1^+) = 0.6, P(e^+ \leftarrow c_2^+) = 0.9, P(e^+ \leftarrow c_3^+) = 0.8,$$

and Fig. 2 (a), $P(e^+ \leftarrow c_1^+, c_2^+, c_3^+) = 0.828$ can be derived by applying Eqn. (1) to the dual gate and Eqn. (2) to the direct gate. Using trees derived from Fig. 2 (a) (by removing some nodes, gates and links), Table 1 is derived.

To derive $P(e^+|c_1^+, c_2^+, c_3^-)$, for instance, reduce Fig. 2 (a) to (b) to obtain $P(e^+ \leftarrow c_1^+, c_2^+) = 0.54$.

TABLE 1
CPT of NIN-AND tree model in Fig. 2 (a)

$P(e^+ c_1^-, c_2^-, c_3^-)$	0	$P(e^+ c_1^+, c_2^-, c_3^+)$	0.92
$P(e^+ c_1^+, c_2^-, c_3^-)$	0.6	$P(e^+ c_1^+, c_2^+, c_3^-)$	0.54
$P(e^+ c_1^-, c_2^+, c_3^-)$	0.9	$P(e^+ c_1^-, c_2^+, c_3^+)$	0.72
$P(e^+ c_1^-, c_2^-, c_3^+)$	0.8	$P(e^+ c_1^+, c_2^+, c_3^+)$	0.828

In Table 1, $P(e^+|c_1^+, c_2^-, c_3^+)$ is larger than both $P(e^+|c_1^+, c_2^-, c_3^-)$ and $P(e^+|c_1^-, c_2^-, c_3^+)$: the consequence of reinforcement between c_1 and c_3 . $P(e^+|c_1^+, c_2^+, c_3^-)$ is less than $P(e^+|c_1^+, c_2^-, c_3^-)$ and $P(e^+|c_1^-, c_2^+, c_3^-)$: the consequence of undermining between the group $\{c_1, c_3\}$ and c_2 .

An NIN-AND tree over e and C , and a set of single-causal probabilities (one for each $c_i \in C$) define an NIN-AND tree model from which a unique CPT is specified. Although a general NIN-AND tree can have a root that involves multiple causes [XJ07], this work focus on NIN-AND trees where each root involves a single cause. Effect and causes in NIN-AND tree models are not limited to binary, but the binary case is the focus of this work. See [Xia12] for NIN-AND tree models over multi-valued variables.

3 CONCISE NIN-AND TREE ENCODING

3.1 Root-Labeled Trees

Multiple NIN-AND trees may encode the same causal model, as illustrated below.

Example 3. From $P(e^+ \leftarrow c_1^+) = 0.95$, $P(e^+ \leftarrow c_2^+) = 0.9$, $P(e^+ \leftarrow c_3^+) = 0.7$, and $P(e^+ \leftarrow c_4^+) = 0.8$, both trees in Fig. 3 yield $P(e^+|c_1^+, c_2^+, c_3^+, c_4^+) = 0.9197$. They encode the same model, where c_1, c_2 and c_3 undermine each other and collectively reinforce c_4 .

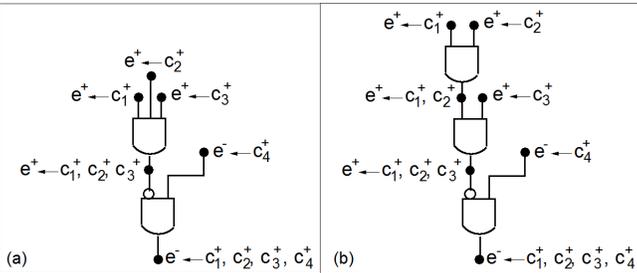


Fig. 3. NIN-AND trees encoding the same model

NIN-AND tree models can be investigated more effectively if each model is encoded by a unique tree topology. We define such trees below.

Definition 3. Let T be an NIN-AND tree. If T contains a gate t that outputs to gate g of the same type (direct

or dual), delete t and direct its inputs to g . Apply such deletions until no longer possible. The resultant NIN-AND tree is minimal.

Every minimal NIN-AND tree T defines an equivalent class, made of all NIN-AND trees that lead to T by Def. 3. They encode the same causal interaction pattern, as shown by Prop. 1.

Proposition 1. Let T be an NIN-AND tree, and T' be its minimal tree. For any set of single-causal probabilities, T and T' yield identical leaf event probability.

Proof: Convert T to T' stepwise according to Def. 3, and we have the sequence $T = T_1, T_2, \dots, T_k = T'$. Consider T_i and T_{i+1} , where inputs of gate t in T_i become inputs of gate g in T_{i+1} with t deleted. For both T_i and T_{i+1} , if t is direct (dual) the contribution of each input of t to the output of g follows Eqn. (2) (Eqn. (1)). Hence, the leaf event probability from T_i is identical to that from T_{i+1} . \square

This equivalence property implies that the space of NIN-AND tree models is equivalently represented by the space of minimal trees. Furthermore, this space can be more compactly represented by a subspace with half of the size, as shown by Prop. 2.

Proposition 2. Let Γ be the collection of minimal NIN-AND trees for n causes with direct leaf gates, and Γ' collects those with dual leaf gates. An one-to-one mapping exists between elements of Γ and Γ' , defined by replacing each gate with the opposite type.

Proof: Prop. 2 follows from a property of minimal trees. If the leaf gate g is direct, then each gate outputting to g is dual, and its inputs are all from direct gates. That is, from the leaf to roots, gates alternate in types. Hence, for every $T \in \Gamma$, there exists a $T' \in \Gamma'$, obtained by changing the type of each gate in T . \square

Prop. 2 allows NIN-AND tree models over n causes to be studied by focusing on one of the two subspaces (Γ and Γ'), whose elements can be equivalently expressed as follows.

Definition 4. Let T be a minimal NIN-AND tree. A root-labeled-tree (RLT) L relative to T is a directed graph obtained by operating on T as follows.

- 1) Delete each gate and direct its inputs to output.
- 2) Delete each non-root label.
- 3) Replace each root label by the name of the corresponding single cause.

Example 4. Fig. 2 (a) is a minimal tree, and its RLT is shown in Fig. 4.

In a RLT, each root corresponds to a single-causal event. Each non-root corresponds to an NIN-AND gate and its output multi-causal event. We therefore may view a non-root from any of its double perspectives. For instance, when a RLT represents a minimal

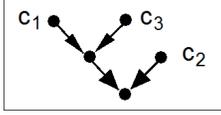


Fig. 4. The RLT of minimal NIN-AND tree in Fig. 2 (a)

tree in Γ , we say that its leaf type is *direct* using the gate perspective. When a RLT represents a tree in Γ' , we say that its leaf is *dual*.

A RLT L may represent a minimal tree in either Γ or Γ' depending on the type of its leaf, as described above. If not mentioned, we assume $L \in \Gamma$ by default, with its leaf being direct.

Example 5. From Fig. 4 and the direct leaf type, Fig. 2 (a) is uniquely recovered.

RLTs encode NIN-AND trees concisely. They provide a basis to effectively study the space of NIN-AND trees as presented below.

3.2 Enumerating Root-Labeled Trees

Enumeration of NIN-AND trees is an important component of our proposed method for NIN-AND tree acquisition. We consider the equivalent task to enumerate RLTs of n roots. Below, we present a new algorithm for the task, by extending a method [Fel04] for counting evolutionary trees.

Let $L_{n,m}$ be a RLT with n roots and $m \in \{1, \dots, n-1\}$ non-roots. Let $\Psi_{n,m}$ be the set of all such RLTs. Denote the set of RLTs with n roots as $\Psi_n = \cup_{m=1}^{n-1} \Psi_{n,m}$.

Algorithm 1. *EnumRLT*(n)

```

1 create  $\Psi_2$ ;
2 for  $i = 3$  to  $n$ , do
3    $\Psi_{i,1} = \text{AddRootToNonRoot}(\Psi_{i-1,1}, c_i)$ ;
4    $\Psi_{i,i-1} = \text{AddRootToArc}(\Psi_{i-1,i-2}, c_i)$ ;
5   for  $m = 2$  to  $i-2$ ,
6      $\Phi = \text{AddRootToNonRoot}(\Psi_{i-1,m}, c_i)$ ;
7      $\Phi' = \text{AddRootToArc}(\Psi_{i-1,m-1}, c_i)$ ;
8      $\Psi_{i,m} = \Phi \cup \Phi'$ ;
9 return  $\Psi_n = \cup_{m=1}^{n-1} \Psi_{n,m}$ ;

```

Line 1 creates $\Psi_2 = \Psi_{2,1}$ that contains a single RLT in Fig. 5 (a). Each iteration of the outer

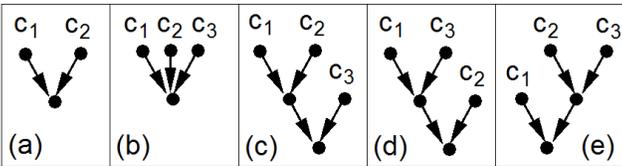


Fig. 5. Illustration of RLT enumeration

for loop enumerates RLTs of i roots in terms of subsets $\Psi_{i,1}, \dots, \Psi_{i,i-1}$. For $i = 3$, line 3 calls *AddRootToNonRoot*($\Psi_{2,1}, c_3$) below to produce $\Psi_{3,1}$ (from $\Psi_{2,1}$) that contains a single RLT in (b).

Procedure 1. *AddRootToNonRoot*($\Psi_{i-1,m}, c_i$)

```

1 initialize  $\Phi = \emptyset$ ;
2 for each RLT  $L \in \Psi_{i-1,m}$ , do
3   for each non-root  $v$  in  $L$ , do
4     create RLT  $L' = L$ ;
5     add root  $c_i$  and arc  $(c_i, v)$  in  $L'$ ;
6     add  $L'$  to  $\Phi$ ;
7 return  $\Phi$ ;

```

Line 4 calls *AddRootToArc*($\Psi_{2,1}, c_3$) below to produce $\Psi_{3,2}$ that contains RLTs in (c), (d), and (e). The RLT in (c) is obtained when v is the leaf, and that in (d) is obtained when $v = c_1$. Lines 5 to 8 of *EnumRLT*(n) are not run for $i = 3$. Hence, Ψ_3 contains four RLTs that are used in the next iteration to produce Ψ_4 .

Procedure 2. *AddRootToArc*($\Psi_{i-1,m-1}, c_i$)

```

1 initialize  $\Phi = \emptyset$ ;
2 for each RLT  $L \in \Psi_{i-1,m-1}$ , do
3   for each node  $v$  in  $L$ , do
4     create RLT  $L' = L$ ;
5     add node  $x$  and arc  $(v, x)$  in  $L'$ ;
6     add root  $c_i$  and arc  $(c_i, x)$  in  $L'$ ;
7     if  $v$  has child  $y$  in  $L$ ,
8       delete arc  $(v, y)$  in  $L'$  and add  $(x, y)$ ;
9     add  $L'$  to  $\Phi$ ;
10 return  $\Phi$ ;

```

Table 2 shows the number of RLTs for $n \leq 7$.

TABLE 2
Number of RLTs for given n

n	2	3	4	5	6	7
No. RLTs	1	4	26	236	2752	39208

4 PAIRWISE CAUSAL INTERACTION

The PCI function presented below forms an important component of one of our proposed methods for NIN-AND tree acquisition.

4.1 PCI Functions of Minimal NIN-AND Trees

An NIN-AND tree T specifies how each pair of causes interact. Such interaction can be formally captured as a function introduced below.

Definition 5. Let x and y be roots of a minimal NIN-AND tree. Let $path_x$ be the directed path from x to the leaf and $path_y$ be that from y . The closest common gate (CCG) of x and y is the first gate common to $path_x$ and $path_y$.

Example 6. In Fig. 2 (a), the CCG of roots $e^- \leftarrow c_1^+$ and $e^- \leftarrow c_3^+$ is dual (top), and that of $e^- \leftarrow c_1^+$ and $e^+ \leftarrow c_2^+$ is direct.

Prop. 3 shows that, given a minimal tree, the interaction between each pair of causes is determined through their CCG.

Proposition 3. Let x and y be roots of a minimal NIN-AND tree, c_i and c_j be their causes, and g be their CCG. If g is direct, then the interaction between c_i and c_j is undermining. Otherwise, it is reinforcing.

Proof: Assume that all causes are false except c_i and c_j . Now all gates, except g , have no more than one active input event and can be removed. The resultant tree has a single gate g with roots x and y . \square

From Prop. 3, a function of pairwise causal interactions is well-defined, as given below.

Definition 6. Given a minimal NIN-AND tree T , the PCI function of T is defined from pairs of distinct causes $\{c_i, c_j\}$ to the set $\{rif, udm\}$, where *rif* stands for reinforcing and *udm* stands for undermining.

Example 7. The PCI function for Fig. 2 (a) is as follows.
 $pci(c_1, c_2) = udm, pci(c_1, c_3) = rif, pci(c_2, c_3) = udm.$

The PCI function of a minimal tree can be obtained from its RLT, plus its leaf gate type.

Example 8. Fig. 4 shows the RLT for Fig. 2 (a). View each non-root in Fig. 4 as a gate. The leaf gate in Fig. 2 (a) is direct. Hence, the leaf in Fig. 4 is direct and its non-root parent is dual. Now the CCG for each pair of roots in Fig. 4 can be determined and so can the PCI function.

Each RLT encodes two alternative minimal trees of opposite leaf gate types. Given the PCI function of one minimal tree, that of the other is easily obtained.

4.2 Expressiveness of PCI Function

Although PCI functions seem simpler than NIN-AND trees, we show that they are sufficiently expressive and uniquely identify their deriving structures. To differentiate minimal NIN-AND trees through their RLTs, we first define isomorphic RLTs recursively in Def. 7 with its notation illustrated in Fig. 6 (a).

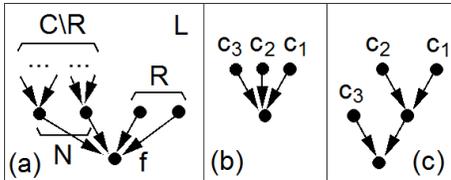


Fig. 6. (a) Illustration of notation in Def. 7. (b), (c) Illustrations of isomorphic RLTs.

Definition 7. Let L be a RLT over a set C of causes and f be its leaf. Denote the set of root parents of f by R and the set of non-root parents of f by N . Let L' be another RLT over C , with f' , R' , and N' defined similarly.

- 1) When $N = \emptyset$, L and L' are isomorphic denoted $L \equiv L'$, iff $N' = \emptyset$.
- 2) When $N \neq \emptyset$, $L \equiv L'$ iff $R = R'$, $|N| = |N'|$, and a one-to-one mapping between $x \in N$ and $x' \in N'$

exists such that the subtree of leaf x and that of leaf x' are isomorphic.

In case (1), since $N = N' = \emptyset$, we have $R = R' = C$.

Example 9. The RLT in Fig. 6 (b) is isomorphic to that in Fig. 5 (b) due to Def. 7 (1). The RLT in Fig. 6 (c) is isomorphic to that in Fig. 5 (c) due to Def. 7 (2).

PCI functions are sufficiently expressive in that whenever $L \neq L'$ their PCI functions differ, and vice versa. This is established in Theorem 1 whose proof utilizes Lemma 1 below.

Lemma 1. Let L and L' be RLTs over C such that no roots v and w sharing a child in L are so in L' , and vice versa. Let a root w be deleted from both RLTs resulting in L_1 and L'_1 , respectively. Then, $L_1 \neq L'_1$.

Fig. 7 illustrates L and L' of Lemma 1 in (a) and (b), and L_1 and L'_1 in (c) and (d).

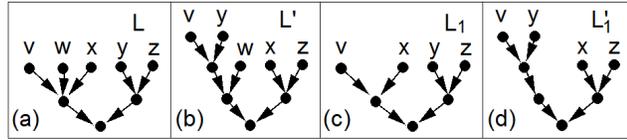


Fig. 7. Illustration of Lemma 1

Proof: We prove by contradiction. Let L and L' be RLTs satisfying the condition of the lemma. Let v and w be roots with a common child in L . Denote the child by c . Since they do not share a child in L' , let y be a root that shares a child with v in L' . Denote the child by d . After the deletion of w from both RLTs, suppose the resultant RLTs satisfy $L_1 \equiv L'_1$.

From Def. 7 (2), the subtree of leaf c in L_1 and the subtree of leaf d in L'_1 are isomorphic because c and d both have parent v . Therefore, the parent set of c in L_1 and that of d in L'_1 must be identical. Since y is the parent of d in L' , it follows that y is the parent of d in L'_1 , y is the parent of c in L_1 , and y is the parent of c in L . Now v and y share a child in both L and L' , which implies that L and L' do not satisfy the condition of the lemma: a contradiction. \square

Theorem 1. Let L and L' be RLTs over a set C of causes, and pci and pci' be their PCI functions, respectively.

Then $L \equiv L'$, iff $pci = pci'$.

Proof: The necessity of the condition is obvious. We prove the sufficiency by induction on $n = |C|$ to show

$$L \neq L' \Rightarrow pci \neq pci'.$$

Without loss of generality, we assume that the leaf type is direct for both RLTs.

For $n = 3$, alternative RLTs are shown in Fig. 5 (b), (c), (d), and (e). If L is as (b) and L' is as (c), then $N = \emptyset$, $N' \neq \emptyset$, and hence $L \neq L'$. We have $pci(c_1, c_2) = udm$ while $pci'(c_1, c_2) = rif$. If L is as (c) and L' is as (d), then $R = \{c_3\}$, $R' = \{c_2\}$, and hence $L \neq L'$. We

have $pci(c_1, c_2) = rif$ while $pci'(c_1, c_2) = udm$. Each other combination of L and L' is similar to one of the two cases above.

Assume $L \not\equiv L' \Rightarrow pci \neq pci'$ for $3 \leq n \leq k$ where $k \geq 3$. We consider $n = k+1$ below. When $N = N' = \emptyset$, we have $L \equiv L'$. Hence, when $L \not\equiv L'$, one of N and N' must be non-empty. We assume $N \neq \emptyset$. By Def. 7, $L \not\equiv L'$ is equivalent to the mutually exclusive and exhaustive cases below.

- 1) $R \neq R'$ and $R \neq \emptyset$
- 2) $R \neq R'$ and $R = \emptyset$
- 3) $R = R' \neq \emptyset$
- 4) $R = R' = \emptyset$, and there exist two roots of a common child in both L and L'
- 5) $R = R' = \emptyset$, and there exist no two roots of a common child in both L and L'

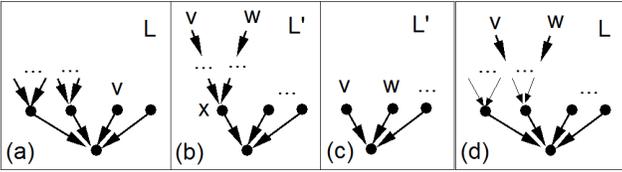


Fig. 8. Illustration of analysis for case (1)

For case (1), $v \in R \setminus R'$ exists as shown in Fig. 8 (a) for L . If $N' \neq \emptyset$, then v is an ancestor of some $x \in N'$ in L' , and x must have another root ancestor $w \in C \setminus R'$ as shown in (b). No matter where w is located in (a), we have $pci(v, w) = udm$ and $pci'(v, w) = rif$.

If $N' = \emptyset$, L' is shown as (c) such that v and w exist in L as (d). We have $pci(v, w) = rif$ and $pci'(v, w) = udm$.

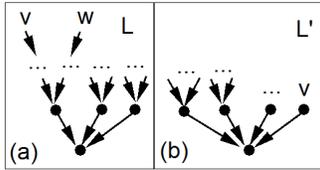


Fig. 9. Illustration of analysis for case (2)

For case (2), v and w exist in L (Fig. 9 (a)) such that v is located in L' as (b). We have $pci(v, w) = rif$ and $pci'(v, w) = udm$ no matter where w is located in L' .

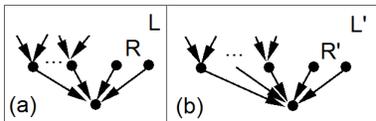


Fig. 10. Illustration of analysis for case (3)

For case (3) (Fig. 10), let L_1 and L'_1 be RLTs obtained by removing roots $R = R'$ from L and L' , respectively. For L_1 and L'_1 , we have $n \leq k$ and $L_1 \not\equiv L'_1$. By inductive assumption, their PCI functions over $C \setminus R$ differ and hence PCI functions for L and L' differ.

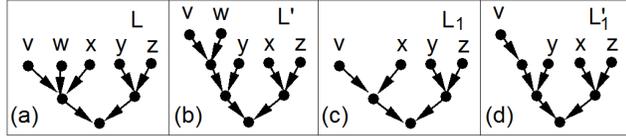


Fig. 11. Illustration of analysis for case (4)

For case (4), e.g., Fig. 11 (a) and (b), v and w share a common child in both L and L' . After w is deleted, we obtain L_1 and L'_1 , e.g., (c) and (d). From $L \not\equiv L'$ and the fact that their non-isomorphism cannot be due to w , it follows $L_1 \not\equiv L'_1$. Since $n = k$ for L_1 and L'_1 , by inductive assumption $L_1 \not\equiv L'_1 \Rightarrow pci_1 \neq pci'_1$, we have $pci_1 \neq pci'_1$. Because for each pair $x, y \in C \setminus \{w\}$, $pci(x, y) = pci_1(x, y)$ holds and $pci'(x, y) = pci'_1(x, y)$ holds, we conclude $pci \neq pci'$.

For case (5), delete a root w from L and L' to get L_1 and L'_1 . By Lemma 1 and an inductive argument similar to case (4), we conclude $pci \neq pci'$. \square

Theorem 1 implies that a minimal NIN-AND tree can be uniquely identified by its PCI function. We will explore this property below.

5 NIN-AND TREE ACQUISITION

Drawing from the results of the last two sections, this section presents two methods for the acquisition of an NIN-AND tree, one based on the structure elimination and another on PCI functions.

5.1 Approach and Assumption

Formally, an NIN-AND tree model M is a tuple $M = (e, C, T, SP)$, where e is an effect, C is the set of all causes of e , T is a minimal NIN-AND tree over e and C , and SP is a set of single-causal probabilities (single-causals), one for each $c_i \in C$. M uniquely specifies a CPT $P(e|C)$. M and $P(e|C)$ are said to be *consistent*.

Furthermore, two models $M = (e, C, T, SP)$ and $M' = (e, C, T', SP')$ are *structurally consistent*, if T and T' are isomorphic. M and M' are said to be *consistent*, if they are consistent with the same CPT.

Given e and C , acquisition of a model M amounts to acquisition of T and SP . SP may be elicited from a domain expert or learned from frequency data. The focus of this paper is on the acquisition of T .

The topology T may be elicited directly from an expert. The task requires the expert to have a good understanding of the syntax and semantics of NIN-AND tree causal models in order to assess and articulate the partial order of interactions among causes and cause groups. This demands a nontrivial amount of training of the expert before elicitation and nontrivial mental exercise of the expert during elicitation.

To ease these burdens for structural acquisition, we explore the following idea to bypass direct tree elicitation. Instead, we acquire (by elicitation or machine

learning) a small number of multi-causal probabilities (multi-causals) in addition to probabilities in SP and generate T from the acquired probabilities. Our investigation is based on the following assumption.

Assumption. Let $P_t(e|C)$ be the true CPT that characterizes the probabilistic relation over e and C such that the following holds.

- 1) There exists an NIN-AND tree model, $M_t = (e, C, T_t, SP_t)$, that is consistent with $P_t(e|C)$.
- 2) All single-causal and some multi-causal probabilities relative to $P_t(e|C)$ can be approximately acquired.

The first condition is justified by the following observation. Reinforcement and undermining capture intuitive patterns of causal interactions, and reinforcement-based causal models, such as the noisy-OR, have been widely applied, e.g., [PPMH94]. The second condition is justified by the knowledge engineering practice in building BNs, where numerical probabilities have been either elicited from experts or learned from available data. Note that the true CPT is never known in practice.

5.2 Structure Elimination (SE) Method

The SE method is based on the RLT enumeration Ψ_n (may be obtained before model acquisition). From Ψ_n , the set TR of minimal NIN-AND trees over C can be obtained by specifying two trees (of opposite leaf types) from each RLT. First, a set SP_a of $n = |C|$ single-causals, e.g., $P_a(e^+ \leftarrow c_i^+)$, is acquired ('a' for 'acquired'). Combining each $T \in TR$ with SP_a , a set NM_a of NIN-AND tree models is obtained. In general, a unique CPT over e and C is defined by each model in NM_a .¹ A set $CPTS_a$ of CPTs is thus defined. Note that there is a one-to-one mapping between TR and NM_a , and generally also between NM_a and $CPTS_a$.

Subsequently, some multi-causals are acquired. Suppose $P_a(e^+ \leftarrow c_i^+, c_j^+, c_k^+)$ is acquired. For each $T' \in TR$, $M' = (e, C, T', SP_a) \in NM_a$ and $P'(e|C) \in CPTS_a$, $P'(e^+ \leftarrow c_i^+, c_j^+, c_k^+)$ is retrieved from $P'(e|C)$. If the difference between $P_a(e^+ \leftarrow c_i^+, c_j^+, c_k^+)$ and $P'(e^+ \leftarrow c_i^+, c_j^+, c_k^+)$ is sufficiently large, $P'(e|C)$ is deemed *inconsistent* with the true CPT, and T' , M' and $P'(e|C)$ are eliminated from TR , NM_a and $CPTS_a$, respectively. Based on such comparisons, all trees in TR except one, T_a , will be eliminated. T_a is the acquired NIN-AND tree and $M_a = (e, C, T_a, SP_a)$ is the acquired model.

Specifically, we adopt the following version of the SE. A set MP_a of K multi-causals are acquired, where K is determined based on expert availability in elicitation or data availability in learning. For each $T' \in TR$, $M' = (e, C, T', SP_a) \in NM_a$, and $P'(e|C) \in CPTS_a$, a corresponding set MP' of multi-causals is retrieved

from $P'(e|C)$. The Euclidean distance between MP_a and MP' is calculated as follows.

$$dm(MP_a, MP') = \sqrt{\frac{1}{K} \sum_{i=1}^K (P_a(e^+ | \underline{x}_i^+) - P'(e^+ | \underline{x}_i^+))^2}. \quad (3)$$

The tree $T_a \in TR$ corresponding to the minimum distance is returned. The algorithm is specified below, where C is a set of n causes of effect e , Ψ_n is the enumeration of RLS of n roots by $EnumRLT(n)$, SP_a is a set of n acquired single-causals, and MP_a is a set of K acquired multi-causals. Although in general any multi-causals may be used in MP_a , we use triple-causals in the remainder of the paper.

Algorithm 2. *TreeAcqBySE*(e, C, Ψ_n, SP_a, MP_a)

Return: a minimal NIN-AND tree;

- 1 $TR = \emptyset$; $NM_a = \emptyset$;
- 2 for each $L \in \Psi_n$,
- 3 set NIN-AND tree T from L with direct leaf;
- 4 set NIN-AND tree T' from L with dual leaf;
- 5 add T and T' to TR ;
- 6 for each $T \in TR$, add $M = (e, C, T, SP_a)$ to NM_a ;
- 7 $T_a = \text{null}$; $mindm = 1$;
- 8 for each $M' = (e, C, T', SP_a) \in NM_a$,
- 9 derive $P'(e|C)$ from M' ;
- 10 get MP' from $P'(e|C)$ corresponding to MP_a ;
- 11 if $dm(MP_a, MP') < mindm$,
- 12 $mindm = dm(MP_a, MP')$; $T_a = T'$;
- 13 return T_a ;

5.3 PCI Function-Based Method

Based on Theorem 1, we propose a PCI function-based method for acquiring a minimal NIN-AND tree as follows. Suppose Ψ_n is given from which the set TR is obtained. From TR , a set $PCIF$ of PCI functions is defined, one for each tree $T \in TR$. First, the set SP_a is acquired as in the SE method. Then, a set DP_a of all double-causals (a total of $n(n-1)/2$ values) is acquired. From SP_a and DP_a , a PCI function $pci_a()$ is derived and compared against each function $pci'() \in PCIF$. If $pci_a()$ matches $pci'()$, then the NIN-AND tree $T' \in TR$ that produces $pci'()$ is the acquired NIN-AND tree and $M_a = (e, C, T', SP_a)$ is the acquired model.

Practical derivation of PCI function

The key operation is the derivation of $pci_a()$ function from SP_a and DP_a . For any pair of causes c_i, c_j in an NIN-AND tree model and their causal probabilities

$$P(e^+ \leftarrow c_i^+), P(e^+ \leftarrow c_j^+), P(e^+ \leftarrow c_i^+, c_j^+),$$

Def. 1 dictates that the PCI function pci must satisfy $pci(c_i, c_j) = rif$ iff

$$P(e^+ \leftarrow c_i^+, c_j^+) \geq \max(P(e^+ \leftarrow c_i^+), P(e^+ \leftarrow c_j^+)), \quad (4)$$

1. When elements of SP_a are not unique, multiple models in NM_a may be consistent.

and $pci(c_i, c_j) = udm$ iff

$$P(e^+ \leftarrow c_i^+, c_j^+) < \min(P(e^+ \leftarrow c_i^+), P(e^+ \leftarrow c_j^+)). \quad (5)$$

In theory, the above decision rule is equivalent to

$$pci(c_i, c_j) = \begin{cases} rif & \text{if } P(e^+ \leftarrow c_i^+, c_j^+) \geq P(e^+ \leftarrow c_i^+) \\ udm & \text{otherwise} \end{cases} \quad (6)$$

and $P(e^+ \leftarrow c_j^+)$ needs not even be involved.

In practice however, due to probability acquisition errors, it is possible that

$$P_a(e^+ \leftarrow c_i^+) < P_a(e^+ \leftarrow c_i^+, c_j^+) < P_a(e^+ \leftarrow c_j^+).$$

Example 10. Suppose that $P_t(e^+ \leftarrow c_i^+) = 0.6$, $P_t(e^+ \leftarrow c_j^+) = 0.9$, and c_i undermines c_j . We have $P_t(e^+ \leftarrow c_i^+, c_j^+) = 0.54$. Acquired probabilities, however, may satisfy

$$P_a(e^+ \leftarrow c_i^+) = 0.56 < P_a(e^+ \leftarrow c_i^+, c_j^+) = 0.59,$$

$$P_a(e^+ \leftarrow c_i^+, c_j^+) = 0.59 < P_a(e^+ \leftarrow c_j^+) = 0.93.$$

Alternatively, suppose c_i reinforces c_j . We have instead $P_t(e^+ \leftarrow c_i^+, c_j^+) = 1 - (0.4 * 0.1) = 0.96$. Acquired probabilities may satisfy

$$P_a(e^+ \leftarrow c_i^+) = 0.56 < P_a(e^+ \leftarrow c_i^+, c_j^+) = 0.91,$$

$$P_a(e^+ \leftarrow c_i^+, c_j^+) = 0.91 < P_a(e^+ \leftarrow c_j^+) = 0.93.$$

In either case, applying Eqn. (6) has a 50% probability of assigning the PCI function value incorrectly. Applying Eqns. (4) and (5) is not even feasible, because both fail.

To address this issue, we develop the decision rule described in Procedure 3.

Procedure 3. $GetPciValue(SP_a, DP_a, c_i, c_j)$

Return: PCI function value $pci_a(c_i, c_j)$;

- 1 get $p_i = P_a(e^+ \leftarrow c_i^+)$ from SP_a ;
- 2 get $p_j = P_a(e^+ \leftarrow c_j^+)$ from SP_a ;
- 3 get $p_{ij} = P_a(e^+ \leftarrow c_i^+, c_j^+)$ from DP_a ;
- 4 if $p_{ij} \geq \max(p_i, p_j)$, return *rif*;
- 5 else if $p_{ij} < \min(p_i, p_j)$, return *udm*;
- 6 $\delta_1 = |p_{ij} - \min(p_i, p_j)|$;
- 7 $\delta_2 = |p_{ij} - \max(p_i, p_j)|$;
- 8 if $\delta_1 < \delta_2$, return *udm*;
- 9 return *rif*;

Normal cases are handled by lines 4 and 5 according to Eqns. (4) and (5). When these equations fail due to probability acquisition errors, the PCI function value is determined by lines 6 to 9. The following example demonstrates correct PCI value assignment when acquisition errors are small.

Example 11. Consider cases in Example 10. For the undermining case, we have $\delta_1 = 0.03$ and $\delta_2 = 0.34$. Hence, $pci_a(c_i, c_j) = udm$ is returned correctly in line 8. For the reinforcing case, we have $\delta_1 = 0.35$ and $\delta_2 = 0.02$. Hence, $pci_a(c_i, c_j) = rif$ is returned correctly in line 9.

Handling invalid PCI functions

To derive a PCI function, $GetPciValue$ should be applied to each cause pair. The number of cause pairs is $n(n-1)/2$. The collection of the $n(n-1)/2$ function values defines a PCI function. Hence, the number of alternative PCI functions for n causes is $2^{n(n-1)/2}$.

TABLE 3
Number of PCI functions and minimal NIN-AND trees

n	No. PCI func.	No. trees	Ratio
3	8	8	1
4	64	52	1.2
5	1024	472	2.2
6	32768	5504	6
7	2097152	78416	26.7
8	268435456	1320064	203
9	68719476736	25637824	2680

Table 3 shows the number of alternative PCI functions and that of minimal NIN-AND trees given n . The right-most column is the ratio between the two.

A PCI function is *valid* if it is derivable from a minimal NIN-AND tree. Otherwise, it is *invalid*. Table 3 shows that beyond $n = 3$, there are more PCI functions than valid ones. It follows that a PCI function derived from $GetPciValue$ may be invalid (due to probability acquisition errors), and there does not exist any corresponding NIN-AND tree.

Let pci and pci' be PCI functions over n causes. We define their distance by counting the number of cause pairs whose function values differ,

$$df(pci, pci') = \sum_{1 \leq i, j \leq n; i \neq j} diff(pci(c_i, c_j), pci'(c_i, c_j)), \quad (7)$$

where the function $diff()$ returns 0 if its arguments are identical and 1 otherwise.

Algorithm 3. $TreeAcqByPCI(e, C, \Psi_n, SP_a, DP_a)$

Return: a minimal NIN-AND tree;

- 1 for each pair $c_i, c_j \in C$,
- 2 $pci_a(c_i, c_j) = GetPciValue(SP_a, DP_a, c_i, c_j)$;
- 3 create the minimal NIN-AND tree set TR from Ψ_n ;
- 4 $T_a = null$; $mindf = n^2$;
- 5 for each $T' \in TR$,
- 6 derive PCI function pci' from T' ;
- 7 if $df(pci_a, pci') = 0$, return T' ;
- 8 if $df(pci_a, pci') < mindf$,
- 9 $mindf = df(pci_a, pci')$; $T_a = T'$;
- 10 return T_a ;

Let pci_a be obtained by $GetPciValue$ and $pci' \in PCIF$. If $df(pci_a, pci') = 0$, pci_a is valid and we return $T' \in TR$ that produces pci' . If $df(pci_a, pci') > 0$ for each $pci' \in PCIF$, then pci_a is invalid. We select a PCI function pci^* , such that $df(pci_a, pci^*)$ is minimal (breaking ties arbitrarily) and return $T^* \in TR$ that produces $pci^*(\cdot)$. The PCI function-based method with the above enhancements is specified in Algorithm 3, where DP_a is a set of acquired double-causals.

6 PARAMETER ACQUISITION ERRORS

This section presents the design of experiments for evaluation of the SE and PCI methods. We discuss potential sources of errors and measures we take to control them.

Experiment Design

The experiments were divided into elicitation-based (EL-based) and machine learning-based (ML-based). First, a true NIN-AND tree model $M_t = (e, C, T_t, SP_t)$ was created from which causal scenarios were generated by Monte Carlo simulations. In EL-based experiments, each human participant was trained into an expert with causal scenarios and then a subset of causal probabilities was elicited. In ML-based experiments, the causal scenarios were used to estimate the causal probabilities.

Once the causal probabilities were acquired, they were used to derive a structure T_a and corresponding model M_a by the SE or PCI method. Effectiveness of the method was measured by the *model acquisition error* $dm(M_a, M_t)$. In the following, we refer to these acquired causal probabilities as *acquired parameters*.

In particular, for scenario simulations, the true CPT $P_t(e|C)$ was derived from M_t . Given a subset $X \subseteq C$ of active causes, a scenario (e, \underline{x}^+) where $e \in \{e^+, e^-\}$ was randomly simulated according to $P_t(e^+ \leftarrow \underline{x}^+)$. For EL-based experiments, after seeing a sufficient number of scenarios, for a sufficient number of distinctive \underline{x}^+ (more below), a participant was deemed to be an expert on M_t . Subsequently, (s)he was asked to articulate the parameters $P_a(e^+ \leftarrow \underline{x}^+)$.

To ensure that a participant's knowledge on M_t is entirely from the training and is not biased by other experiences, we created M_t to be about phenomena in an imaginary planet, where atmospheric conditions are influenced by conditions of the moon, clouds, certain insects, and plants. An Environment Simulator (ES) was implemented accordingly to allow (s)he to specify active causes \underline{x}^+ and observe simulated effects e in terms of picture icons. This setup ensures the condition (1) in the Assumption.

It is important to note that training experts according to a generated model M_t is more advantageous than using real experts from some application environment. The mental model of a real expert is not directly accessible and hence acquisition errors cannot be accurately measured. On the other hand, training experts according to a generated M_t allows convenient error measurement as shown below.

Parameter acquisition errors

In the ML-based experiments, $P_a(e^+ \leftarrow \underline{x}^+)$ is estimated from the relative frequency

$$F(e^+ \leftarrow \underline{x}^+) = N(e^+ \leftarrow \underline{x}^+)/N(\underline{x}^+),$$

where $N(e^+ \leftarrow \underline{x}^+)$ is the number of occurrences of scenario (e^+, \underline{x}^+) and $N(\underline{x}^+)$ is that for \underline{x}^+ . Since

$N(\underline{x}^+)$ is finite, the parameter acquisition suffers from *sampling errors*:

$$P_a(e^+ \leftarrow \underline{x}^+) = F(e^+ \leftarrow \underline{x}^+) \neq P_t(e^+ \leftarrow \underline{x}^+).$$

In the EL-based experiments, since each participant forms $P_a(e^+ \leftarrow \underline{x}^+)$ from the training experience on $F(e^+ \leftarrow \underline{x}^+)$, the parameter acquisition also suffers from the sampling errors.

Furthermore, in the EL-based experiments, a participant may not be able to retain and articulate either $N(e^+ \leftarrow \underline{x}^+)$ and $N(\underline{x}^+)$, or $F(e^+ \leftarrow \underline{x}^+)$ accurately [KST82]. We refer to these additional errors as *retention-articulation (RA) errors*.

Controlling parameter acquisition errors

Without adequately controlling the parameter acquisition errors, the effectiveness of SE and PCI methods cannot be accurately evaluated. To control the sampling errors, we analyze the number $\eta = N(\underline{x}^+)$ required for $F(e^+ \leftarrow \underline{x}^+)$ to be sufficiently close to $p = P_t(e^+ \leftarrow \underline{x}^+)$.

Scenarios (e, \underline{x}^+) are simulated through a Bernoulli process. Denote $\theta = N(e^+ \leftarrow \underline{x}^+) \in \{0, 1, 2, \dots, \eta\}$ and it has a binomial distribution

$$P(\theta) = C(\eta, \theta)p^\theta(1-p)^{\eta-\theta},$$

where $C(\eta, \theta)$ denotes the number of θ -combinations from η elements. Now $F(e^+ \leftarrow \underline{x}^+) = p$ iff $\theta = \eta p$. Define a *threshold* $\gamma \in (0, 1)$. $F(e^+ \leftarrow \underline{x}^+)$ is deemed close to p iff $|F(e^+ \leftarrow \underline{x}^+) - p| \leq \gamma$, which is equivalent to

$$|\eta F(e^+ \leftarrow \underline{x}^+) - \eta p| = |\theta - \eta p| \leq \eta \gamma.$$

Hence, $P(|\theta - \eta p| \leq \eta \gamma)$ signifies the chance with which $F(e^+ \leftarrow \underline{x}^+)$ is close to p .

TABLE 4

$P(|\theta - \eta p| \leq \eta \gamma)$ for $\eta = 100$ and $\gamma = 0.1$

p	0.1/0.9	0.2/0.8	0.3/0.7	0.4/0.6	0.5
$P()$	0.999	0.992	0.979	0.968	0.965

Table 4 lists $P(|\theta - \eta p| \leq \eta \gamma)$ for $p = 0.1, 0.2, \dots, 0.9$ when $\eta = 100$ and $\gamma = 0.1$. Therefore, if each \underline{x}^+ is simulated $\eta = 100$ times, the probability by which $F(e^+ \leftarrow \underline{x}^+)$ differs from $P_t(e^+ \leftarrow \underline{x}^+)$ by no more than 0.1 will be at least 0.965 no matter what value $P_t(e^+ \leftarrow \underline{x}^+)$ is.

Based on the analysis, we simulated each \underline{x}^+ for 100 scenarios in ML-based experiments. For EL-based experiments, we implemented the ES so that a participant was required to observe at least 100 scenarios for each distinct training \underline{x}^+ before the elicitation starts.

To control RA errors, for each distinct training \underline{x}^+ , the pair of relative frequencies $F(e^+ \leftarrow \underline{x}^+)$ and $F(e^- \leftarrow \underline{x}^+)$ experienced by the participant during training was made available during elicitation and was shown in a stacked bar graph (Fig. 12). This

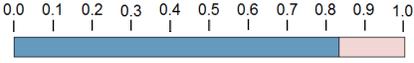


Fig. 12. Stacked bar graph for $F(e^+ \leftarrow \underline{x}^+) = 0.83$

visual hint for the experienced $F(e^+ \leftarrow \underline{x}^+)$ essentially eliminates retention errors. While it reduces articulation errors, it does not eliminate them as it is visual whereas $P_a(e^+ \leftarrow \underline{x}^+)$ is elicited numerically.

The stacked bar graph also allows us to identify outliers in acquired parameters from careless or disinterested participants. For each $P_a(e^+ \leftarrow \underline{x}^+)$, its articulation error can be calculated from the corresponding frequency in the ES log. Since the stacked bar graph bounds articulation errors to below 0.1, any error larger than 0.1 is detected as an outlier.

7 EXPERIMENTAL EVALUATION

The objective of experiments was to evaluate whether the SE and PCI methods effectively acquire NIN-AND trees, under the Assumption (Section 5.1). In addition to EL-based and ML-based experiments, we also compared with the Chow-Liu method [CL68].

7.1 Elicitation-Based Experiments

EL-based experimental setup

Participants were recruited from computer science and engineering students (second year or above). Each participant was trained with a distinct true model $M_t = (e, C, T_t, SP_t)$, where $n = 4$. T_t was randomly selected from 52 alternatives (see Table 3) and SP_t was randomly generated. A training session lasted 10 minutes, during which a participant selected \underline{x}^+ (in the ES) to observe its causal scenarios according to the required parameters and the minimum number (100) of scenarios.

To facilitate evaluation, the SE and PCI methods were compared with the *direct numerical* (DN) method, where each causal probability in $P_t(e|C)$ was directly elicited. The total number of parameters is $2^4 - 1 = 15$, where -1 is due to $P_t(e^+ | c_1^-, \dots, c_4^-) = 0$ (Section 2). For PCI, we elicited 10 parameters (4 single-causals and $C(4, 2) = 6$ double-causals). For SE, we elicited 8 parameters (4 single-causals and $C(4, 3) = 4$ triple-causals) so that the parameters differ from PCI.

Each data set consists of a number of causal probabilities elicited from one participant. A data set for evaluation of DN, SE, or PCI method contains 15, 8, or 10 elicited probabilities, respectively. If any acquired parameter was found to be an outlier (Section 6), the entire dataset was dropped from further analysis. After removing 3 outlier datasets, the number of data sets collected are 23, 29, 29, respectively.

The distance between sets of probabilities was measured by Euclidean distance similar to Eqn. (3). If eight parameters were elicited from a participant,

his or her parameter acquisition error was calculated using the eight and their counterparts in $P_t(e|C)$. For a model acquired by any method, its acquisition error was calculated using the 15 parameters of the acquired model and their counterparts in $P_t(e|C)$.

EL-based experimental results

Parameter acquisition errors consist of sampling and RA errors. For each participant, these errors were computed from the true CPT $P_t(e|C)$, a set $F(e|C)$ of relative frequencies of simulated scenarios in the ES log, and a set $P_a(e|C)$ of probability parameters directly elicited. $F(e|C)$ and $P_a(e|C)$ from a given participant may contain less than 15 parameters, when only the PCI or SE method was applied. In particular, for each participant, the sampling error was obtained by $dm(P_t(e|C), F(e|C))$, the RA error was obtained by $dm(F(e|C), P_a(e|C))$, and the parameter acquisition error was by $dm(P_t(e|C), P_a(e|C))$.

The three types of errors over all participants are summarized in Table 5. As shown, the elicitation aid by stacked bar graphs had an effective control of RA errors. As a result, parameter acquisition errors are mainly composed of sampling errors.

TABLE 5

Sample means ($\hat{\mu}$) and standard deviations ($\hat{\sigma}$) of parameter acquisition errors over all participants

	Sampling Err.	RA Err.	Para. Acq. Err.
$\hat{\mu}$	0.0293	0.0076	0.0301
$\hat{\sigma}$	0.0096	0.0038	0.0099

The DN method directly elicits the CPT $P_a^{DN}(e|C)$, which is referred to as the *CPT elicited by DN*. For the SE method, a structure T_a (by TreeAcqBySE) and a model M_a are derived from elicited probabilities and $P_a^{SE}(e|C)$ (the *CPT elicited by SE*) is defined from M_a . Similarly, $P_a^{PCI}(e|C)$ (the *CPT elicited by PCI*) through TreeAcqByPCI is defined.

For each data set, the Euclidean distance between the true CPT and the CPT elicited by each method was calculated. For each method, DN, SE, and PCI, the distances from corresponding data sets are summarized in Table 6. Out of the 29 data sets applicable

TABLE 6

Sample means ($\hat{\mu}$) and standard deviations ($\hat{\sigma}$) of model acquisition errors by DN, SE and PCI methods

	DN	SE	PCI
$\hat{\mu}$	0.0301	0.0356	0.0281
$\hat{\sigma}$	0.0099	0.0343	0.0146

to the PCI method, the true NIN-AND tree topology was recovered in 28 of them (recovery rate 96.6%). Out of the 29 data sets applicable to SE, the true topology was recovered in 26 (recovery rate 89.7%).

To compare the three methods, the Friedman test (3 columns, 23 rows) was performed, based on results from 23 data sets (the remaining data set each contains only parameters sufficient for one of PCI and SE). A rank 1 is associated with the lowest distance. The rank sums for DN, PCI, and SE are 50.0, 42.5, and 45.5, respectively, and the test statistic is 1.239. Hence, model acquisition errors from the three methods are comparable (the null hypothesis is not rejected). Since the PCI method acquired 10 parameters, SE acquired 8, while DN acquired all 15, the results demonstrate that SE and PCI methods improve efficiency in model acquisition, while maintaining comparable accuracy.

7.2 Machine Learning-Based Experiments

ML-based experimental setup

Although EL-based experiments provided useful empirical evaluations, they were tedious and time consuming due to its human-based nature. The size of our datasets was limited by the participants that we could recruit. The true models used were limited to $n = 4$ to reduce the session length for participants. During these experiments, it became clear that, the inputs needed to evaluate the SE and PCI methods are the acquired parameters, and it does not matter whether they are elicited from humans or estimated from data. Hence, ML-based experiments are also valid as long as parameter acquisition errors are comparable. They are advantageous because the feasibility for larger n values and sample sizes, which allows more accurate measure of relative merits of the methods.

ML-based experiments were conducted for 4,000 simulated true models. They were divided into four groups with $n = 4, 5, 6,$ and $7,$ respectively. Each group consists of 1,000 true models. For each true model $M_t = (e, C, T_t, SP_t)$, T_t was randomly selected (e.g., for $n = 7,$ from 78416 enumerated alternatives) and SP_t was randomly generated. Each T_t has between 1 and $n - 1$ non-roots. Each method of DN, PCI and SE was applied to each model in each group.

For each true model M_t , the relative frequency $F(e^+ \leftarrow \underline{x}^+)$ was simulated from $P_t(e|C)$ for each subset $X \subseteq C$ of active causes. We have chosen the number η of simulated scenarios to be 100 to make sampling errors comparable with parameter acquisition errors in EL-based experiments. The acquired parameter $P_a(e^+ \leftarrow \underline{x}^+)$ was then estimated by $F(e^+ \leftarrow \underline{x}^+)$. A total of $2^n - 1$ parameters were acquired for each M_t .

The CPT $P_a^{DN}(e|C)$ acquired by DN is defined directly by the $2^n - 1$ acquired parameters. For the PCI method, the input is a proper subset of the above $2^n - 1$ acquired parameters: n single causals and $C(n, 2)$ double-causals (a total of $n * (n + 1)/2$). The output is the acquired model CPT $P_a^{PCI}(e|C)$ (consisting of $2^n - 1$ parameters). For the SE method, the input is a proper subset of n single causals and

$C(n, 3)$ triple causals, and the acquired model CPT is $P_a^{SE}(e|C)$. Table 7 summarizes the number of acquired parameters by each method.

TABLE 7
Number of input parameters by DN, SE and PCI

n	# probs for DN	# probs for SE	# probs for PCI
4	15	8	10
5	31	15	15
6	63	26	21
7	127	42	28

ML-based experimental results

Table 8 summarizes ML-based experimental results. For the DN method, sample means and standard deviations of $dm(P_t(e|C), P_a^{DN}(e|C))$ for each group is shown as $\hat{\mu}$ -dm and $\hat{\sigma}$ -dm in column 2. Corresponding results for PCI and SE are shown in columns 3 and 5. For the PCI method, the true topology recovery rate Rcv and the typical model learning time tt (by a 2.9 GHz laptop) are shown in column 4. Corresponding results for SE are shown in column 6.

TABLE 8
Summary of ML-based experimental result

n	DN		PCI		SE	
	$\hat{\mu}$ -dm	$\hat{\sigma}$ -dm	$\hat{\mu}$ -dm	Rcv (%)	$\hat{\mu}$ -dm	Rcv (%)
4	0.0373	0.0101	0.0328	98.7	0.0486	80.5
			0.0163	<0.001	0.0429	<0.001
5	0.0365	0.0084	0.0316	97.7	0.0348	89.2
			0.0152	<0.001	0.0198	0.016
6	0.0356	0.0077	0.0297	97.8	0.0308	89.8
			0.0141	0.015	0.0139	0.41
7	0.0360	0.0071	0.0301	96.6	0.0300	89.8
			0.0145	0.1	0.0130	14.3

Since $dm(P_t(e|C), P_a^{DN}(e|C))$ is equivalent to the parameter acquisition error, results in column 2 indicate the magnitudes of parameter acquisition errors for all three methods. They are comparable to that of EL-based experiments (Table 5).

To compare model acquisition errors of the three methods, the Friedman test (3 columns, 1,000 rows) was performed for each group. The test statistics for $n = 4, 5, 6,$ and 7 are 159.9, 256.8, 364.4, and 418.2, respectively. Hence, the null hypothesis (no significant difference between the three methods) is rejected for each group at the level of significance $\alpha = 0.01$.

For the post-hoc analysis, rank sums for each group and each method are shown in Table 9. The rank sum differences for three pairs of methods are derived in Table 10. With an one-sided test for each of the three pairs, the result is the following. PCI is significantly better than DN for $n = 4, 5, 6, 7$ at $\alpha = 0.01$. SE is significantly better than DN for $n = 4, 5, 6, 7$ at $\alpha = 0.01$. PCI is significantly better than SE for $n = 4, 5, 6,$

TABLE 9
Rank sums of model acquisition errors

n	RS^{DN}	RS^{PCI}	RS^{SE}
4	2280.0	1714.5	2005.5
5	2409.0	1741.5	1849.5
6	2489.0	1702.0	1809.0
7	2527.0	1708.5	1764.5

TABLE 10
Rank sum differences for post-hoc analysis

n	$RS^{DN} - RS^{PCI}$	$RS^{DN} - RS^{SE}$	$RS^{SE} - RS^{PCI}$
4	565.5	274.5	291.0
5	667.5	559.5	108.0
6	787.0	680.0	107.0
7	818.5	762.5	56.0

at $\alpha = 0.01$, but for $n = 7$, it is only better than SE at $\alpha = 0.11$. We attribute the improved performance of SE (relative to PCI) for $n = 7$ to the use of 50% more parameters than PCI (42 versus 28).

The superior performance of PCI over SE is also reflected by its true topology recovery rate Rcv (columns 4 and 6 in Table 8). The Rcv from PCI is at least 6% higher than SE, for each n .

The analysis shows that PCI and SE perform better than DN, and PCI is the best among the three. The PCI and SE methods also used fewer number of acquired parameters (Table 7). The number of acquired parameters by DN is $O(2^n)$, that by SE is $O(n^3)$, and that by PCI is $O(n^2)$. The performance of the PCI method makes it particularly attractive, as it uses the least number of acquired parameters (except for $n = 4$) but has the best model acquisition accuracy and topology recovery rate.

7.3 On True Model Based Approach

Our experiments extended empirical evaluations that were previously performed for causal models [ZD04], [MD08] by including ML-based experiments. Previous methods compared the elicited model CPT and the observed relative frequency parameters (*frequency-based*), while we compared the acquired model with the true model (*true-model-based*).

To evaluate the relative merit of the true-model-based and frequency-based evaluations, the Friedman test (3 columns, 1,000 rows) was performed to compare $dm(P_t(e|C), P_a^{DN}(e|C))$, $dm(P_t(e|C), P_a^{PCI}(e|C))$ (true-model-based), and $dm(P_a^{DN}(e|C), P_a^{PCI}(e|C))$ (frequency-based), with rank sums for each group shown in Table 11.

The Friedman test statistics for $n = 4, 5, 6, 7$ are much large than the critical value at $\alpha = 0.01$. Hence, the null hypothesis is rejected for each n . In the post-hoc analysis, one-sided pairwise tests show that, for each n , the true-model-based PCI model acquisition error is significantly better than DN, but

TABLE 11
True-model and frequency-based rank sums for PCI

n	RS^{DN}	RS^{PCI}	RS^{PCI-f}
4	2102.0	1557.0	2341.0
5	1977.0	1392.0	2631.0
6	1857.0	1255.0	2888.0
7	1794.0	1234.0	2972.0

the frequency-based error is significantly worse than DN. For instance, when $n = 7$, the difference between RS^{DN} and RS^{PCI} is 560, and that between RS^{PCI-f} and RS^{DN} is 1178, both exceeding the critical value 104.07 at $\alpha = 0.01$.

Similar analysis was also performed for the SE method, with the rank sums shown in Table 12. Again,

TABLE 12
True-model and frequency-based rank sums for SE

n	RS^{DN}	RS^{SE}	RS^{SE-f}
4	1926.0	1717.0	2357.0
5	1930.0	1447.0	2623.0
6	1817.0	1304.0	2879.0
7	1776.0	1250.0	2974.0

the null hypothesis is rejected for each n . Pairwise tests in the post-hoc analysis show that the true-model-based SE model acquisition error is significantly better than DN, but the frequency-based error is significantly worse than DN.

These observations suggest that the true-model-based approach is more accurate, while the frequency-based evaluation may lead to biased conclusions.

7.4 Comparing PCI with Chow-Liu Tree Method

To illustrate the fundamental difference between the acquisition of NIN-AND trees and that of BN DAGs, the PCI method was compared against the well-known C-L method [CL68].

Given a set of discrete variables, the C-L method learns a tree structured BN. To apply C-L in the context of NIN-AND tree acquisition, we assume that the set of variables is the BN family $Ce = C \cup \{e\}$, where causes in C are marginally independent of each other (the typical case in an NIN-AND tree model).

To compare PCI with C-L, we generate a true NIN-AND tree model, sample it, and acquire an NIN-AND tree and a C-L tree from the frequency data. To apply PCI, a set of single-causals and double-causals is to be estimated. To apply C-L, however, a set of double-variable marginal probabilities is to be estimated. Note that neither set can be derived from the other set. This means that PCI and C-L must be applied to different sets of input data, which generally contain different levels of sampling errors. This makes it difficult for us to make a fair comparison between the performance of PCI and that of C-L.

To make a fair comparison, we assume that the frequency data are perfect, without sampling errors. That is, input probabilities to each method are directly defined by the true joint probability distribution (JPD) $P(e, C)$ (the subscript t is removed for simplicity). In particular, the input to PCI consists of $P(e \leftarrow c_i)$ for $i = 1, \dots, n$, and $P(e \leftarrow c_i, c_j)$ for each pair of i and $j \neq i$. The input to C-L consists of $P(x, y)$ for each pair of x and $y \neq x$, where $x, y \in Ce$.

The C-L method identifies a tree skeleton (of undirected links) as a maximum spanning tree, where the weight of a link is defined as the mutual information between the pair of variables. If we apply C-L to the family Ce , no link between any pair of causes is included in the tree, due to the marginal independence (and hence zero link weight). It follows that the skeleton must be a star, with e at the center.

Next, the skeleton is directed. The C-L requires that each variable has a single parent except for the root. Pearl [Pea88] extended C-L trees to polytrees, where a variable may have multiple parents, by directing a V structure, $x \rightarrow z \leftarrow y$, such that x and y are independent and the other two pairs are dependent. If Pearl’s method is applied to the above star, e will be the common child of all causes, whose associated numerical distribution is $P(e|C)$. As the result, we have made no progress in reducing its space complexity.

We therefore direct links of the star by following the original C-L method: allowing no more than one parent per variable. The root r can be arbitrarily chosen, and be associated with $P(r)$. Each other variable x with parent y is associated with $P(x|y)$. Since the JPD of the resultant BN is independent of the choice of root, we choose e as the root. The JPD of the C-L tree BN is $P^{CL}(e, C) = P(e)P(c_1|e)\dots P(c_n|e)$, whose corresponding causal CPT is

$$P^{CL}(e|C) = P(e)P(c_1|e)\dots P(c_n|e)/(P(c_1)\dots P(c_n)). \quad (8)$$

To compare PCI with C-L, we compare $dm(P(e|C), P^{PCI}(e|C))$ with $dm(P(e|C), P^{CL}(e|C))$. Since applying the PCI method to the perfect data always returns the true model, it follows that $dm(P(e|C), P^{PCI}(e|C)) = 0$. Hence, the relative merit of PCI versus C-L is completely characterized by $dm(P(e|C), P^{CL}(e|C))$. Table 13 shows the experimental result over 4,000 randomly simulated true model JPDs (each consisting of an NIN-AND tree model plus a marginal for each cause), with 1,000 true models for each n value. The result

TABLE 13

Summary of model acquisition errors by C-L method

n	4	5	6	7
$\hat{\mu}$ -dm	0.204	0.171	0.150	0.135
$\hat{\sigma}$ -dm	0.020	0.018	0.020	0.022

indicates that the causal CPT acquired by the C-L

method has on average at least 0.13 absolute error per conditional probability, given perfect input data. Since the corresponding error by the PCI method is zero, the PCI method is clearly superior. Note that the acquired C-L BN encodes that any c_i and c_j , where $i \neq j$, are independent given e . Since Ce is a BN family, this is false. The NIN-AND tree acquired by PCI makes no such false encoding.

The experiment illustrates the fundamental difference between BN DAG learning and NIN-AND tree acquisition. Both are techniques for reducing the 2^n parameter complexity. DAG learning does so by encoding conditional independence. Since there is none within a BN family, when forced to do so, C-L encodes false conditional independence relations and results in an inaccurate model. NIN-AND tree acquisition does so by encoding causal interaction and causal independence within the BN family, and therefore is able to reduce the number of parameters from 2^n to n with an accurate model.

8 CONCLUSION

NIN-AND tree causal models are based on intuitive notions of reinforcing and undermining interactions, among causes. Since they are expressive while requiring only a linear number of parameters, they provide a powerful tool for modeling local distributions in graphical models (e.g., BNs). This paper meets a major challenge in the NIN-AND tree modeling: the structural acquisition.

The contributions of this work include the compact RLT representation of NIN-AND trees, a direct RLT enumeration algorithm, the formal establishment of the expressiveness of PCI functions, and the PCI and SE algorithms for NIN-AND tree acquisition through the acquisition of a small set of low order causal probabilities. Both methods are shown to have significantly better model acquisition accuracy than the direct method, while requiring $O(n^2)$ (by PCI) or $O(n^3)$ (by SE) acquired parameters. Both methods are applicable to elicitation-based and machine learning-based acquisitions.

We decomposed parameter acquisition errors into sampling and RA (in elicitation) errors. RA errors may be reduced through training and/or technical aids. For sampling errors, we have shown that, about 100 scenarios per causal combination is sufficient for our methods to work well.

The PCI method extracts the PCI function from acquired single and double-causal probabilities. Alternatively, the function may be directly elicited from the expert by asking for each pair of causal interaction, saving the acquisition of double-causal probabilities. Using such qualitative information in BN construction has been previously studied [DG95], [RG02]. This variation, however, is not applicable to machine learning-based model acquisition.

Although the number of NIN-AND trees grows super-exponentially in n (faster than $n!$), it is not an insurmountable obstacle for many reasons. First, RLTs of n roots can be enumerated offline once and reused for acquisition of multiple models. Second, although the number of variables in a BN is unbounded, because a NIN-AND tree models a single CPT (over a single family of variables) in a BN and BNs are intended to be sparse, n is not unbounded. It is expected to be small and commonly no more than 10. Third, when n becomes large, techniques such as divorcing [JN07] can be applied to split the family before applying NIN-AND tree modelling to each reduced family. Finally, the PCI and SE methods, as presented, have time complexities that are linear on the number of NIN-AND trees, and hence are also super-exponential. However, algorithms that only examine a subspace of NIN-AND trees but are based on the fundamental ideas of PCI and SE are possible.

This paper assumes an unknown true NIN-AND tree model. Future work may relax this assumption and explore NIN-AND tree modeling for arbitrary local distributions. In addition, we have proposed SE and PCI methods in the context of binary variables. Another avenue of future research would be to extend this method to allow for multi-valued variables. We have measured modeling errors using the Euclidean distance. Alternative measures, e.g., KL divergence, may also be applied and compared. In our experiments, all acquired parameters assumed roughly the same levels of acquisition errors. It is conceivable that such errors may grow larger in practice as the number of active causes involved in a parameter grows. In that case, comparisons of model acquisition errors are expected to be more favorable for the SE and PCI methods. This is because they need to acquire probabilities over no more than a few active causes (three in our experiments), while the DN method must acquire probabilities over all active causes. More refined experiments may verify such a conjecture.

ACKNOWLEDGMENT

Support from NSERC is acknowledged. We thank J. Zhu, D. Stanley, and B. Nonnecke for assistance.

REFERENCES

- [CBS08] R. Chang, W. Brauer, and M. Stetter. Modeling semantics of inconsistent qualitative knowledge for quantitative bayesian network inference. *Neural Networks*, 21(2-3):182–192, 2008.
- [CL68] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, (14):462–467, 1968.
- [DG95] M.J. Druzdzel and L.C. Van Der Gaag. Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In *Proc. 11th Conf. Uncertainty in Artificial Intelligence*, pages 141–148. Morgan Kaufmann, 1995.
- [Fel04] J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Mass., 2004.

- [GD00] S.F. Galan and F.J. Diez. Modeling dynamic causal interaction with Bayesian networks: temporal noisy gates. In *Proc. 2nd Inter. Workshop on Causal Networks*, pages 1–5, 2000.
- [HB96] D. Heckerman and J.S. Breese. Causal independence for probabilistic assessment and inference using Bayesian networks. *IEEE Trans. on System, Man and Cybernetics*, 26(6):826–831, 1996.
- [JN07] F.V. Jensen and T.D. Nielsen. *Bayesian Networks and Decision Graphs (2nd Ed.)*. Springer, New York, 2007.
- [KST82] D. Kahneman, P. Slovic, and A. Tversky, editors. *Judgment under uncertainty: heuristics and biases*. Cambridge University Press, 1982.
- [LG04] J.F. Lemmer and D.E. Gossink. Recursive noisy OR - a rule for estimating complex probabilistic interactions. *IEEE Trans. on System, Man and Cybernetics, Part B*, 34(6):2252–2261, 2004.
- [MD08] P.P. Maasant and M.J. Druzdzel. An independence of causal interactions model for opposing influences. In M. Jaeger and T.D. Nielsen, editors, *Proc. 4th European Workshop on Probabilistic Graphical Models*, pages 185–192, Hirtshals, Denmark, 2008.
- [NCF12] M. Neil, X. Chen, and N. Fenton. Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization. *IEEE Trans. Knowledge and Data Engineering*, 24(7):1306–1312, 2012.
- [NFN00] M. Neil, N. Fenton, and L. Nielsen. Building large-scale bayesian networks. *Knowledge Eng. Rev.*, 15(3):257–284, 2000.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [PPMH94] M. Pradhan, G. Provan, B. Middleton, and M. Henrion. Knowledge engineering for large belief networks. In *Proc. 10th Conf. Uncertainty in Artificial Intelligence*, pages 484–490, Seattle, Washington, 1994.
- [RG02] S. Renooij and L.C. Van Der Gaag. From qualitative to quantitative probabilistic networks. In *Proc. 18th Conf. Uncertainty in Artificial Intelligence*, pages 422–429. Morgan Kaufmann, 2002.
- [Xia12] Y. Xiang. Non-impeding noisy-and tree causal models over multi-valued variables. *International J. Approximate Reasoning*, 53:988–1002, 2012.
- [XJ07] Y. Xiang and N. Jia. Modeling causal reinforcement and undermining for efficient CPT elicitation. *IEEE Trans. Knowledge and Data Engineering*, 19(12):1708–1718, 2007.
- [XLZ09] Y. Xiang, Y. Li, and J. Zhu. Towards effective elicitation of NIN-AND tree causal models. In L. Godo and A. Pugliese, editors, *Inter. Conf. on Scalable Uncertainty Management (SUM 2009)*, LNCS 5785, pages 282–296. Springer-Verlag Berlin Heidelberg, 2009.
- [XTZ+11] Y. Xiang, M. Truong, J. Zhu, D. Stanley, and B. Nonnecke. Indirect elicitation of NIN-AND trees in causal model acquisition. In S. Benferhat and J. Grant, editors, *Inter. Conf. on Scalable Uncertainty Management (SUM 2011)*, LNCS 6929, pages 261–274. Springer-Verlag Berlin Heidelberg, 2011.
- [XZL09] Y. Xiang, J. Zhu, and Y. Li. Enumerating unlabeled and root labeled trees for causal model acquisition. In Y. Gao and N. Japkowicz, editors, *Advances in Artificial Intelligence, LNAI 5549*, pages 158–170. Springer, 2009.
- [ZD04] A. Zagorecki and M.J. Druzdzel. An empirical study of probability elicitation under Noisy-OR assumption. In *Proc. 17th Inter. Florida Artificial Intelligence Research Society Conf.*, pages 880–885, 2004.