Modeling Causal Reinforcement And Undermining With Noisy-AND Trees

Y. Xiang and N. Jia

University of Guelph, Canada

Abstract. Causal modeling, such as noisy-OR, reduces probability parameters to be acquired in constructing a Bayesian network. Multiple causes can reinforce each other in producing the effect or can undermine the impact of each other. Most existing causal models do not consider their interactions from the perspective of reinforcement or undermining. We show that none of them can represent both interactions. We present the first explicit causal model that can encode both reinforcement and undermining and we show how to use such a model to support efficient probability elicitation.

1 Introduction

A Bayesian network (BN) [7] encodes concisely the probabilistic knowledge about a large problem domain. However, when a variable has a large number of parent variables in the BN, acquisition of the corresponding conditional probability table (CPT) is exponential on the number of parents.

Pearl pioneered idea of noisy-OR model [7]. Henrion [5] added to noisy-OR model the leaky probability. Diez [1] and Srinivas [9] extended noisy-OR from binary to multi-valued variables. Heckerman and Breese [4] analyzed a collection of causal independence relations that allows efficient acquisition of conditional probability tables in BNs. Lemmer and Gossink [6] proposed recently the recursive noisy-OR model.

When multiple causes are present, they can reinforce each other in producing the effect or they can undermine the impact of each other. Unlike [6], previous work do not consider causal interactions among variables from the perspective of reinforcement or undermining, and model parameters are limited to probabilities of single cause events. All previously proposed causal models, including noisy-OR, recursive noisy-OR, noisy-MAX, noisy-AND and noisy-addition, are limited to represent either reinforcement or undermining but not both.

In this work, we present an noisy-AND tree model that represents arbitrary causal interactions among a set of causes, some of them are reinforcing and others are undermining. Reinforcement and undermining are encoded explicitly to support probability elicitation and probabilities for multi-cause events can be incorporated as model parameters if so desired.

In Section 2, we introduce the terminology and define formally reinforcement and undermining. Section 3 presents how reinforcement and undermining can be

modeled uniformly using noisy-AND gates. Section 4 proposes the noisy-AND tree model and how to use it to obtain causal probability is described in Section 5. We present, in Section 6, how to use noisy-AND trees to model causal interaction when default independence assumptions do not hold. We demonstrate elicitation of CPTs with noisy-AND trees in Section 7. Section 8 compares related causal models with noisy-AND trees.

2 Background

We aim to assess a conditional probability distribution of a variable x conditioned on a set of variables Y based on their causal relation. The causes that we consider are uncertain causes. Following Lemmer and Gossink [6], an *uncertain cause* is a cause that can produce an effect but does not always do so. We denote a set of binary cause variables as $X = \{c_1, ..., c_n\}$ and their effect variable (binary) as e. For each c_i, we denote $c_i = true$ by c_{i1} and $c_i = false$ by c_{i0}. Similarly, we denote $e = true$ by e_1 and $e = false$ by e_0.

We refer to the event that a cause c_i causes an effect e to occur as a *causal event*. We denote this causal event by $e_1 \leftarrow \{c_{i1}\}$ or simply $e_1 \leftarrow c_{i1}$, and we denote its negation that c_i does not cause e as $e_1 \nleftarrow c_{i1}$. Note that causal event $e_1 \leftarrow c_{i1}$ is not just the concurrence of c_{i1} and e_1. With the above notation, c_i is an uncertain cause of e if and only if $0 < P(e_1 \leftarrow c_{i1}) < 1$.

We denote causal event that a set of causes $X = \{c_1, ..., c_n\}$ causes e as $e_1 \leftarrow \{c_{11}, ..., c_{n1}\}$, or simply $e_1 \leftarrow c_{11}, ..., c_{n1}$ or $e_1 \leftarrow \mathbf{x_1}$. When the cause set is indexed, such as $W_i = \{c_1, ..., c_n\}$, the causal event may be denoted $e_1 \leftarrow \mathbf{w_{i1}}$. We allow broad interpretations of causal event by a set of causes, as will be seen in later sections. For instance, we are not limited to the interpretation in [6]: *the effect is caused by at least one of the causes.*

Pearl [7] regards a cause as a certain cause, whose occurrence always causes the effect. He encodes the causal uncertainty using an uncertain inhibitor. The conjunction of a certain cause and an inhibitor in his formulation is equivalent to an uncertain cause.

When modeling a domain with a BN, the set of all causes of an effect variable e is its parents. We denote the set of all causes of e by C. To capture causes that we do not wish to represent explicitly, we include a leaky cause variable in C (as one of c_1 through c_n).

Probability of causal event can be used to assess CPT $P(e|C)$. For example, if $C = \{c_1, c_2, c_3, c_4\}$, then $P(e_1|c_{11}, c_{20}, c_{31}, c_{41}) = P(e_1 \leftarrow c_{11}, c_{31}, c_{41})$. Note that only cause variables of value *true* are included in the right-hand side of the causal probability.

When multiple causes are present, they may reinforce each other in producing the effect. That is, their combined influence is greater than that from only some of them. Alternatively, multiple causes may undermine each other in producing the effect. Below, we define reinforcement and undermining formally.

Definition 1 *Let $R = \{W_1, W_2, ...\}$ be a partition of a set X of causes, R' be a proper subset of R, and Y be the union of elements in R'. Sets of causes in R*

are said to `reinforce` each other, if for every subset $R' \subset R$, it holds that

$$P(e_1 \leftarrow \mathbf{y_1}) \leq P(e_1 \leftarrow \mathbf{x_1}).$$

Otherwise, sets of causes in R are said to `undermine` each other.

When each R_i is a singleton, *reinforcement* corresponds to *positive causality* in [6] and *undermining* corresponds to *inhibition*. Hence, reinforcement and undermining are more general. They allow modeling of reinforcement of sets of causes when causes in some set are undermining. Similarly, they allow modeling of undermining of sets of causes when causes in some set are reinforcing. This will become more clear in Section 4.

3 Noisy-AND Gates for Reinforcement and Undermining

We propose to model reinforcement as well as undermining uniformly based on AND gate, which we refer to as *noisy-AND gate*. It builds on previous work with noisy-OR [7] and noisy-AND [2], but takes a different perspective from reinforcing and undermining interactions among uncertain causes.

We assume that, *by default*, sets of reinforcing causes $R = \{W_1, ..., W_m\}$, where W_i and W_j are disjoint for all i and j, satisfy *failure conjunction*:

$$(e_1 \nleftarrow \mathbf{w_{11}}, ..., \mathbf{w_{m1}}) = (e_1 \nleftarrow \mathbf{w_{11}}) \wedge ... \wedge (e_1 \nleftarrow \mathbf{w_{m1}}). \tag{1}$$

That is, sets of reinforcing causes fail to produce effect if each set of causes has failed to produce the effect. We also assume that, *by default*, sets of reinforcing causes satisfy *failure independence*:

$$P(e_1 \nleftarrow \mathbf{w_{11}}, ..., \mathbf{w_{m1}}) = P(e_1 \nleftarrow \mathbf{w_{11}}) \, ... \, P(e_1 \nleftarrow \mathbf{w_{m1}}). \tag{2}$$

That is, failure events $e_1 \nleftarrow \mathbf{w_{11}}$, ..., $e_1 \nleftarrow \mathbf{w_{m1}}$ are independent of each other.
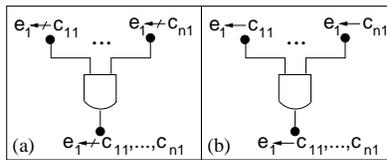
Fig. 1. Noisy-AND gate.

We model the default reinforcing interaction graphically with the noisy-AND gate in Figure 1 (a), where each $W_i = \{c_i\}$ is a singleton, $m = n$, failure conjunction is expressed by the AND gate, and failure independence is expressed by lack of direct connection between individual failure events. The following Lemma confirms their reinforcement. Due to space limit, we omit proofs for all formal results.

3

Lemma 1 *Let $R = \{W_1, W_2, ...\}$ be a partition of a set X of uncertain causes of effect e and sets in R satisfy Eqns (1) and (2). Then, interaction among sets of causes in R is reinforcing.*

When each W_i is a singleton, Eqn (2) can be alternatively written as

$$P(e_1 \leftarrow c_{11}, ..., c_{n1}) = 1 - \prod_{i=1}^{n}(1 - P(e_1 \leftarrow c_{i1})), \qquad (3)$$

which is the noisy-OR model [7]. Therefore, Lemma 1 also formalizes relation between noisy-OR and reinforcement. We refer to the noisy-AND gate in Figure 1 (a) as the default model for reinforcement. The default model represents only one possible reinforcement among sets of causes. We present representation for different reinforcements in Section 6.

Next, we consider undermining. We assume that, *by default*, sets of undermining causes satisfy *success conjunction*:

$$e_1 \leftarrow \mathbf{x_1} = (e_1 \leftarrow \mathbf{w_{11}}) \wedge ... \wedge (e_1 \leftarrow \mathbf{w_{m1}}). \qquad (4)$$

That is, when sets of undermining causes succeed in causing the effect *in undermining way*, each set of causes must have been effective. We emphasize that the success occurs in an undermining way. If any set of causes has occurred but has failed to be effective, it would not undermine the other sets of causes. We also assume that, *by default*, sets of undermining causes *succeed independently*, i.e.,

$$P(e_1 \leftarrow \mathbf{x_1}) = P(e_1 \leftarrow \mathbf{w_{11}}) ... P(e_1 \leftarrow \mathbf{w_{m1}}). \qquad (5)$$

The following lemma confirms their undermining interaction, whose proof is straightforward.

Lemma 2 *Let $R = \{W_1, W_2, ...\}$ be a partition of a set X of uncertain causes of effect e and sets in R satisfy Eqns (4) and (5). Then, interaction among sets of causes in R is undermining.*

Again, the default model represents only one possible undermining interaction among sets of causes. We describe representation of other undermining interactions in Section 6.

4 Noisy-AND Trees

Consider two sets X and Y of causes that reinforce each other. It is possible that causes within X undermine each other, and so do causes within Y. In general, such interplay of causal interactions of different natures can form a hierarchy. In this section, we present a graphical representation to model such a hierarchy. It is based on noisy-AND gates and has a tree topology. We term it *noisy-AND tree*. We assume that a domain expert is comfortable to assess reinforcing and

undermining interactions among causes according to some partial order and is able to articulate the hierarchy.

For example, consider a patient in the process to recover from a disease D. Taking medicine M helps recovery and so does regular exercise. Patient's normal diet contains minerals that facilitate recovery but taking with medicine M reduces effectiveness of both.

The causes and effect involved are defined as follows:

- e_1 : Recovery from disease D within a particular time period.
- c_{11}: Taking medicine M.
- c_{21}: Regular exercise.
- c_{31}: Patient takes his/her normal diet.

For the purpose of prognosis, one needs to assess $P(e_1 \leftarrow c_{11}, c_{21}, c_{31})$. To ease the task, a physician may consider first undermining interaction between c_1 and c_3. (S)he then considers reinforcing interaction between sets $\{c_1, c_3\}$ and $\{c_2\}$. Thus, the physician has articulated an order for stepwise assessment. In addition, the physician also assesses

$$P(e_1 \leftarrow c_{11}) = 0.85, P(e_1 \leftarrow c_{21}) = 0.8, P(e_1 \leftarrow c_{31}) = 0.7.$$

If this is all the information that the physician can provide, the causal interaction can be modeled as the noisy-AND tree in Figure 2 (a).
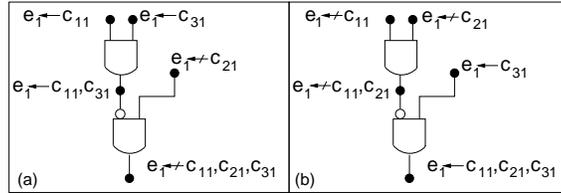
Fig. 2. (a) Noisy-AND tree model of disease example. (b) Alternative model.

From the upper AND gate and Eqn (5), we derive $P(e_1 \leftarrow c_{11}, c_{31}) = 0.595$, an effect of undermining. The output of the upper AND gate is negated (shown by the white oval) before entering the lower AND gate and the corresponding event has probability $P(e_1 \nleftarrow c_{11}, c_{31}) = 0.405$. From the lower AND gate and Eqn (2), we derive

$$P(e_1 \nleftarrow c_{11}, c_{21}, c_{31}) = P(e_1 \nleftarrow c_{11}, c_{31})P(e_1 \nleftarrow c_{21}) = 0.081,$$

and $P(e_1 \leftarrow c_{11}, c_{21}, c_{31}) = 0.919$. The following defines a noisy-AND tree in general.

Definition 2 *Let e be an effect and $X = \{c_1, ..., c_n\}$ be a set of uncertain causes that is known to have occurred. An* `noisy-AND tree` *for modeling causal interaction among elements of X is a directed tree where the following holds:*

1. *There are two types of nodes on the tree. An* `event` *node is shown as a black oval and a* `gate` *node is shown as an AND gate. Each event node has an in-degree ≤ 1 and an out-degree ≤ 1. Each gate has an in-degree ≥ 2 and an out-degree 1.*
2. *Every link connects an event node with a gate node. There are two type of links:* `forward` *links and* `negation` *links. Each link is directed from its* `tail` *node to its* `head` *node consistently along the input-to-output stream of gates. A forward link is shown as a line and is implicitly directed. A negation link is shown as a line with a white oval at the head and is explicitly directed.*
3. *All terminal nodes are event nodes and each is labeled by a causal event in the form $e_1 \leftarrow \mathbf{y}$ or $e_1 \nleftarrow \mathbf{y}$. Exactly one terminal node, called the* `leaf`, *is connected to the output of a gate and has $\mathbf{y} = \mathbf{x_1}$. Each other terminal node is connected to the input of a gate and is a* `root`. *For each root, $\mathbf{y}$ is a proper subset of $\mathbf{x_1}$, it holds $\bigcup_i \mathbf{y_i} = \mathbf{x_1}$ with i indexing roots, and for every two roots with $\mathbf{y_j}$ and $\mathbf{y_k}$, it holds $\mathbf{y_j} \cap \mathbf{y_k} = \emptyset$.*
4. *Multiple inputs of a gate g must be in one of the following cases:*
   (a) *Each is either connected by a forward link to a node labeled with $e_1 \leftarrow \mathbf{y}$, or by a negation link to a node labeled with $e_1 \nleftarrow \mathbf{y}$. Output of g is connected by a forward link to a node labeled with $e_1 \leftarrow \cup_i \mathbf{y_i}$.*
   (b) *Each is either connected by a forward link to a node labeled with $e_1 \nleftarrow \mathbf{y}$, or by a negation link to a node labeled with $e_1 \leftarrow \mathbf{y}$. Output of g is connected by a forward link to a node labeled with $e_1 \nleftarrow \cup_i \mathbf{y_i}$.*

Degree restriction in Condition 1 ensures that an event represents the output of no more than one gate and is connected to the input of no more than one gate. Condition 4 ensures that inputs to each gate either all corresponds to causal events in the form of $e_1 \leftarrow \mathbf{y}$, or all corresponds to causal events in the form of $e_1 \nleftarrow \mathbf{y}$. Semantically, 4 (a) corresponds to undermining sets of causes and 4 (b) corresponds to reinforcing sets.

5   Noisy-AND Tree Evaluation

A noisy-AND tree can be used to evaluate $P(e_1 \leftarrow \mathbf{x_1})$ given $P(e_1 \leftarrow \mathbf{y})$ or $P(e_1 \nleftarrow \mathbf{y})$ for each root node. The computation can be performed recursively by decomposing the noisy-AND tree into subtrees. The following lemma shows that such decomposition is valid.

Lemma 3 *Let T be a noisy-AND tree, the leaf of T be v, and the gate connected to v be g. Let v and g be deleted from T, as well as the links incoming to g. In the remaining graph, each component is either an isolated event node or a noisy-AND tree.*

A noisy-AND tree can be evaluated according to the following algorithm.

Algorithm 1 *GetCausalEventProb(T)*
Input: A noisy-AND tree T.

denote leaf of T by v and gate connected to v by g;
for each node w directly connected to input of g, do
   if probability $P(w)$ for event at w is not specified,
     denote sub-AND-tree with w as the leaf by T_w;
     $P(w) = GetCausalEventProb(T_w)$;
  if (w, g) is a forward link, $P'(w) = P(w)$;
  else $P'(w) = 1 - P(w)$;
$P(v) = \prod_w P'(w)$;
return $P(v)$;

The following theorem establishes soundness of GetCausalEventProb. We define the *depth* of a noisy-AND tree to be the maximum number of gate nodes contained in a path from a root to the leaf.

Theorem 1 *Let T be a noisy-AND tree where probability for each root node is specified in the range $(0, 1)$ and $P(v)$ be returned by $GetCausalEventProb(T)$. Then $P(v)$ is a probability in the range $(0, 1)$ and it combines given probabilities according to reinforcement or undermining specified by the topology of T.*

Note that the topology of T is a crucial piece of knowledge. For the above example, suppose the physician articulates a different order, which is shown in Figure 2 (b). The physician feels that reinforcing interaction between c_1 and c_2 should be considered first. The undermining interaction between sets $\{c_1, c_2\}$ and $\{c_3\}$ should then be considered. Applying GetCausalEventProb, we obtain $P(e_1 \nleftarrow c_{11}, c_{21}) = 0.03$ and $P(e_1 \leftarrow c_{11}, c_{21}, c_{31}) = 0.679$.

6 Relaxing Default Assumptions

A noisy-AND tree assumes, by default, failure independence for reinforcing sets of causes and success independence for undermining sets of causes. For given sets of causes, the expert may disagree with such assumptions. This may manifest in terms of disagreement of the expert with output event probability of a noisy-AND gate. When this occurs, noisy-AND tree representation allows easy modification by deleting the corresponding AND gate from the tree. In particular, let g be the gate in question and its output be connected to node v. If the expert disagrees with the event probability computed for node v, the entire subtree with v as the leaf can be discarded by deleting the link (g, v). Node v remains in the resultant new noisy-AND tree as a root node. The expert can then specify a proper event probability for v.

For instance, with the noisy-AND tree in Figure 2 (a), suppose that the expert disagrees with $P(e_1 \leftarrow c_{11}, c_{31}) = 0.595$. Instead, (s)he feels that 0.4 is more appropriate. Note that this assignment is consistent with the undermining interaction between c_1 and c_3, but the degree of undermining is different from what the default assumption dictates. We can then remove root nodes labeled by $e_1 \leftarrow c_{11}$ and $e_1 \leftarrow c_{31}$ as well as the gate that they are connected to. As the result, node $e_1 \leftarrow c_{11}, c_{31}$ becomes a root node and $P(e_1 \leftarrow c_{11}, c_{31}) = 0.4$ can

be assigned to it. Applying GetCausalEventProb to the new noisy-AND tree, we obtain $P(e_1 \leftarrow c_{11}, c_{21}, c_{31}) = 0.88$.

This flexibility of noisy-AND tree allows it to be used in an interactive way, increasing its expressive power as a tool for probability elicitation: An expert can start by articulating a noisy-AND tree where each root is labeled by a single cause c_i. The default assumptions on failure and success independence now allow computation of probability for each non-root causal event. This can be viewed as the first approximation of the expert's subjective belief. The expert can then examine each computed event probability and decide if it is consistent with his/her belief.

Upon identification of disagreement over a node v connected to the output of a gate g, the expert can trace backward to input events connected to g. The expert will decide whether (s)he disagrees with the probabilities of any input events. If no such disagreement is identified, then the expert must be disagreeing with the degree of reinforcement or undermining implied by the assumption on failure or success independence. (S)he can then assess a probability for the output event as we illustrated above. Note that this assessment, with the computed probability as reference, is easier than an assessment to be made from vacuum. On the other hand, if disagreement with the probability of an input event is identified, the processing continues by tracing further back towards root nodes.

It is possible that as the expert traces disagreements, makes modifications to event probabilities, and deletes subtrees, a deep noisy-AND tree started with becomes very shallow in the end. Many root node labels now consist of a subset of causes, instead of a single one at the start. The topology of the resultant noisy-AND tree becomes very different. This does not mean that the original noisy-AND tree was wrong. It has disappeared after serving its useful role in elicitation.

7 Elicitation of CPTs With Noisy-AND Trees

We demonstrate how to use noisy-AND trees to elicit CPTs in BNs with an example. Consider an effect (child) variable e with a set of seven causes (parents) in a BN: $c_1, ..., c_7$. Suppose that a domain expert identifies the following three subsets of causes and interaction within each subset:

- Subset s_1: c_1 and c_2 are undermining each other.
- Subset s_2: c_2, c_3 and c_4 are reinforcing each other.
- Subset s_3: c_6 and c_7 are reinforcing each other.

The expert assesses that interaction between subsets s_1 and s_2 is also undermining and, together as a group, they reinforce s_3. Without further quantitative information, these assessments produce the noisy-AND tree in Figure 3 (a). Suppose that the following probabilities for single-cause events are also provided:

$$P(e_1 \leftarrow c_{11}) = 0.65, P(e_1 \leftarrow c_{21}) = 0.35, P(e_1 \leftarrow c_{31}) = 0.8,$$

$$P(e_1 \leftarrow c_{41}) = 0.3, P(e_1 \leftarrow c_{51}) = 0.6, P(e_1 \leftarrow c_{61}) = 0.75, P(e_1 \leftarrow c_{71}) = 0.55.$$
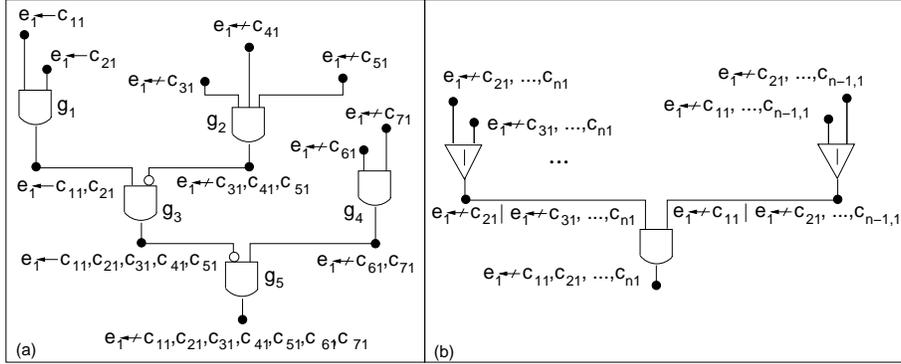
Fig. 3. (a) An example noisy-AND tree. (b) Graphical model for recursive noisy-OR.

To assess $P(e_1|c_{11}, ..., c_{71})$, we apply GetCausalEventProb to obtain

$$P(e_1|c_{11}, ..., c_{71}) = P(e_1 \leftarrow c_{11}, ..., c_{71}) = 0.912.$$

To assess $P(e_1|c_{11}, c_{21}, c_{30}, c_{41}, c_{51}, c_{61}, c_{71})$, eliminate node $e_1 \nleftarrow c_{31}$ from Figure 3 (a) and modify output labels for g_2, g_3 and g_5. The evaluation gives

$$P(e_1|c_{11}, c_{21}, c_{30}, c_{41}, c_{51}, c_{61}, c_{71}) = P(e_1 \leftarrow c_{11}, c_{21}, c_{41}, c_{51}, c_{61}, c_{71}) = 0.906.$$

We have used the same noisy-AND tree to assess both probabilities above. This is not necessary. That is, noisy-AND trees do not require that different causal probabilities to be assessed using the same tree. If the expert feels that a particular combination of a subset of causes follows a different pattern of interaction, a distinct noisy-AND tree can be used, without producing invalid CPT. Commonly, we expect that one tree can be used for assessment of all probabilities in a CPT. If the expert is happy with the result, the complexity of his/her assessment task is only $O(n)$, where n is the number of causes.

Suppose that the expert believes that 0.906 is too high for $P(e_1|c_{11}, c_{21}, c_{30}, c_{41}, c_{51}, c_{61}, c_{71})$ and (s)he attributes to the output from gate g_4 $P(e_1 \nleftarrow c_{61}, c_{71}) = 0.113$ as too low. Instead, (s)he believes 0.2 is a better assessment. In response, we remove the subtree with g_4 as the leaf and specify 0.2 as the probability for the new root event node $e_1 \nleftarrow c_{61}, c_{71}$. GetCausalEventProb now generates $P(e_1|c_{11}, c_{21}, c_{30}, c_{41}, c_{51}, c_{61}, c_{71}) = 0.833$.

8 Related Models Of Causal Interaction

We compare noisy-AND trees with related models of causal interaction. As we have defined reinforcement and undermining under the binary context, the following analysis is restricted to such context if appropriate.

Some models of causal interaction are limited to represent either reinforcement or undermining but not both. Noisy-MAX model [1] becomes noisy-OR model when variables are binary. Therefore, from Lemma 1, when domain is binary, noisy-MAX represents only reinforcing interaction.

Similarly, noisy-MIN model [2] becomes noisy-AND when variables are binary. Hence, according to Lemma 2, when domain is binary, noisy-MIN represents only undermining interaction.

Lemmer and Gossink [6] proposed RNOR to model reinforcement. To assess effect probability due to a set of causes, RNOR model can combine causal probabilities due to subsets of causes, where each subset may not be singleton. Their combination at subset level has influenced our thinking in formulation of noisy-AND trees. According to RNOR, for a set of causes $X = \{c_1, ..., c_n\}$, if $P(e_1 \leftarrow c_{11}, ..., c_{n1})$ is not provided by the expert, it is estimated as

$$P(e_1 \leftarrow c_{11}, ..., c_{n1}) = 1 - \prod_{i=1}^{n} \frac{1 - P(e_1 \leftarrow c_{11}, ..., c_{i-1,1}, c_{i+1,1}, ..., c_{n1})}{1 - P(e_1 \leftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1})} \quad (6)$$

as long as causes in X are reinforcing. However, if causes in X are undermining, the result from the equation may not be a valid probability.

No graphical representation of RNOR was proposed in [6]. We present a graphical model which reveals the independence assumption underlying RNOR. Using failure events, we rewrite Eqn (6) below:

$$P(e_1 \nleftarrow c_{11}, ..., c_{n1}) = \prod_{i=1}^{n} \frac{P(e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+1,1}, ..., c_{n1})}{P(e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1})} \quad (7)$$

$$= \prod_{i=1}^{n} \frac{P((e_1 \nleftarrow c_{i+1,1}) \wedge (e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1}))}{P(e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1})} \quad (8)$$

$$= \prod_{i=1}^{n} P(e_1 \nleftarrow c_{i+1,1} | e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1}) \quad (9)$$

Figure 3 (b) shows the graphical model of RNOR based on Eqn (7) and Eqn (9). A gate representing "conditioning" has been introduced and is shown as a triangle with a vertical bar in the center. We refer to the gate as a COND gate. The output of a COND gate is the event of its left input event conditioned on its right input event. Note that $e_1 \nleftarrow c_{i+1,1} | e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1}$ is a well defined event. Each input event to a COND gate is associated with a real *potential*. Its output event is assigned a potential defined by the division of the two input potentials (the one in the left divided by that in the right). For the AND gate, its output event is assigned a potential defined by the product of potentials of its inputs. Inputs of each gate are not connected in any path other than through the gate.

Eqn (9) and Figure 3 (b) reveal that RNOR model assumes that conditional failure event denoted by $e_1 \nleftarrow c_{i+1,1} | e_1 \nleftarrow c_{11}, ..., c_{i-1,1}, c_{i+2,1}, ..., c_{n1}$ (where i runs from 1 to n) is independent of each other. This is not surprising as RNOR is derived from rewriting Eqn (3) and it assumes failure independence among all causes. However, when RNOR is used recursively by replacing default probabilities on input of some COND gates, the independence assumption is invalidated,

10

while the topology of the graphical model and the rule of probability combination (Eqn (6)) remain and do not reflect such invalidation.

On the other hand, independence assumptions made in noisy-AND trees are local to each gate. Assumption made relative to a gate governs only the probability combination at the output of the gate and is independent of the assumptions made at other gates. When the default probability produced by a gate is replaced and the corresponding subtree removed, it does not invalidate any independence assumptions at other gates in the remaining noisy-AND tree. That is, modification of a noisy-AND tree does not invalidate the coherence of the underlying independence assumptions.

Noisy-addition [3] can represent neither reinforcement nor undermining. Consider a noisy-adder with two binary causes c_1 and c_2 whose domains are $\{0, 1\}$. It has the following DAG model, where i_1 and i_2 are intermediate variables and effect $e = i_1 + i_2$:

$$c_1 \longrightarrow i_1 \longrightarrow e \longleftarrow i_2 \longleftarrow c_2$$

The model assumes $P(i_j = 0 | c_j = 0) = 1$ and $0 < P(i_j = 1 | c_j = 1) < 1$ for $j = 1, 2$. For simplicity, we assume $P(i_1 = 1 | c_1 = 1) = P(i_2 = 1 | c_2 = 1)$ and denote their value by q. Note that $P(e = 1 | c_1 = 1) = P(i_1 = 1 | c_1 = 1)$. To decide whether this model can represent reinforcement or undermining, $P(e = 1 | c_1 = 1, c_2 = 1)$ should be compared with q. We derive the following:

$$P(e = 1 | c_1 = 1, c_2 = 1)$$
$$= P(i_1 = 0, i_2 = 1 | c_1 = 1, c_2 = 1) + P(i_1 = 1, i_2 = 0 | c_1 = 1, c_2 = 1)$$
$$= P(i_1 = 0 | c_1 = 1)P(i_2 = 1 | c_2 = 1) + P(i_1 = 1 | c_1 = 1)P(i_2 = 0 | c_2 = 1)$$

Denoting $P(e = 1 | c_1 = 1, c_2 = 1)$ by r, we have $r = 2q(1-q)$. If $q < 0.5$, then $r > q$. If $q > 0.5$, then $r < q$. By Definition 1, if a causal model is reinforcing, then no matter what value $P(e_1 \leftarrow \mathbf{y_1})$ is, the relation $P(e_1 \leftarrow \mathbf{y_1}) \leq P(e_1 \leftarrow \mathbf{x_1})$ must hold and reverse of the inequality must hold for undermining. Being unable to maintain the inequality across the entire range of values for $P(e_1 \leftarrow \mathbf{y_1})$ implies that noisy-addition is unable to represent either reinforcement or undermining.

Noisy-AND trees differ from those considered by Heckerman and Breese [4] in that the amechanistic model has essentially a star topology and other three models (decomposable, multiply decomposable and temporal) are essentially binary trees. When the binary tree is instantiated according to noisy-OR, noisy-AND, noisy-MAX, noisy-MIN, noisy-addition, it inherits limitations of these models as discussed above. In these models, each root node must be a single cause variable, while noisy-AND trees allow a root node to represent a causal event of multiple causes.

Pearl [8] analyzed causation using functional causal models. Our work is consistent with his functional approach and in particular proposes noisy-AND trees as a useful boolean functional model.

9   Conclusions

Causal interactions may be reinforcing or undermining. Distinction of them can facilitate causal modeling and CPT elicitation in construction of Bayesian networks. We have shown that existing causal models can model either one type of interactions (such as noisy-OR, noisy-AND, noisy-MAX, noisy-MIN and recursive noisy-OR) or none of them (such as noisy-addition). We present the first explicit causal model, termed noisy-AND trees, that can encode both reinforcement and undermining. Furthermore, existing causal models, except recursive noisy-OR, limit model parameters to probabilities of single cause events, and recursive noisy-OR introduces inconsistent dependence assumptions when probabilities of multi-cause events are integrated through recursion. On the other hand, noisy-AND trees integrate probabilities of both single cause events and multi-cause events coherently. Therefore, noisy-AND trees provide a simple yet powerful new approach for knowledge elicitation in probabilistic graphical models.

Acknowledgements

References

1. F.J. Diez. Parameter adjustment in Bayes networks: The generalized noisy or-gate. In D. Heckerman and A. Mamdani, editors, *Proc. 9th Conf. on Uncertainty in Artificial Intelligence*, pages 99–105. Morgan Kaufmann, 1993.
2. S.F. Galan and F.J. Diez. Modeling dynamic causal interactiosn with Bayesian networks: temporal noisy gates. In *Proc. 2nd Inter. Workshop on Causal Networks*, pages 1–5, 2000.
3. D. Heckerman. Causal independence for knowledge acquisition and inference. In D. Heckerman and A. Mamdani, editors, *Proc. 9th Conf. on Uncertainty in Artificial Intelligence*, pages 122–127. Morgan Kaufmann, 1993.
4. 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.
5. M. Henrion. Some practical issues in constructing belief networks. In L.N. Kanal, T.S. Levitt, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 161–173. Elsevier Science Publishers, 1989.
6. J.F. Lemmer and D.E. Gossink. Recursive noisy OR - a rule for estimating complex probabilistic interactions. *IEEE SMC, Part B*, 34(6):2252–2261, 2004.
7. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
8. J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.
9. S. Srinivas. A generalization of noisy-or model. In D. Heckerman and A. Mamdani, editors, *Proc. 9th Conf. on Uncertainty in Artificial Intelligence*, pages 208–215. Morgan Kaufmann, 1993.