This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2891680, IEEE Transactions on Knowledge and Data Engineering

An extension to the *noisy-OR* function to resolve the 'explaining away' deficiency for practical Bayesian network problems

Norman E. Fenton, Takao Noguchi, and Martin Neil

Abstract—The "leaky noisy-OR" function is a common and popular method used to simplify the elicitation of complex conditional probability tables in Bayesian networks involving Boolean variables. It has proven to be useful for approximating the required relationship in many real-world situations where there is a set of two or more variables that are potential causes of a single effect variable. However, one of the properties of leaky noisy-OR is Conditional Inter-causal Independence (CII). This property means that the 'explaining away' behaviour – one of the most powerful benefits of BN inference – is not present when the effect variable is observed as false. Yet, for many real-world problems where the leaky noisy-OR has been considered, this behaviour would be expected, meaning that leaky noisy-OR is deficient as an approximation of the required relationship in such cases. There have been previous attempts to adapt the noisy-OR to resolve this problem. However, they require too many additional parameters to be elicited. We describe a simple but powerful extension to leaky noisy-OR that requires only a single additional parameter. While it does not solve the CII problem in all cases, it does resolve most of the explaining away deficiencies that occur in practice. The problem and solution is illustrated using an example from intelligence analysis.

Index Terms— Knowledge Representation Formalisms and Methods, Probabilistic algorithms, Decision support

1 INTRODUCTION

Causal probabilistic networks, also known as Bayesian networks (BNs), are a well-established graphical formalism for encoding conditional probabilistic relationships between uncertain variables. The nodes of a BN represent variables and the arcs represent causal or influential relationships between them. Probabilistic inference in BNs is based on belief propagation [1].

One of the most formidable practical challenges in building BN models for decision support and risk assessment is to define the necessary conditional probability tables (CPTs). When data is sparse we must rely on judgment from domain experts for this. However, even when a node y and all its n parents are Boolean, the size of the CPT grows exponentially with n; experts must provide 2^n probability value parameters – one for each of the parent state combinations. This is known to be practically intractable and highly error-prone even for n as small as 3 [2]–[4]. To partly address this challenge modellers have identified generic BN structures [5] in which fewer parameters are necessary to capture the required set of conditional dependencies. A particularly common example is the ***noisy-OR*** function and its extension the **leaky noisy-OR**, which, as the name suggests, are like the Boolean OR operator but with noise added (all formal definitions are provided in Section 3).

The leaky noisy-OR requires only $n + 1$ parameters for the full CPT specification, and is a good approximation of a wide range of BN model fragments. Hence, it has proven extremely popular and useful in practice [6]–[12], especially as it is implemented in commercial BN tools [13], [14]. The assumptions underlying the leaky noisy-OR generally work well for forward inference (predicting the effect of any combination of parent observations on y). However, for backward inference the results may not be what modellers expected. In particular, when y is observed to be false the normal 'explaining away' behaviour – one of the most powerful benefits of BN inference – fails: after observing the state of any parent the remaining parents become independent. Formally, this is because leaky noisy-OR has an inherent property called conditional inter-causal independence (CII) which blocks the normal explaining away behaviour in cases where y is false. The ramifications of this property are not widely known among practitioners, which is in itself a problem that hopefully is addressed by this paper.

In Section 2 we review the history of noisy-OR and related work. In Section 3 we formally define noisy-OR and discuss its benefits and limitations, using an example drawn from intelligence analysis. In Section 4 we discuss the ramifications of the CII property and explain how to 'break it'. However, previous proposed solutions require, in the worst case, the elicitation of an exponential number of additional parameters, and so these solutions are not practically useful. Our novel and important contribution, described in Section 5, is a very simple but powerful extension to the leaky noisy-OR that requires only a single additional parameter. While this single parameter solution does not solve the CII problem in all cases, it does resolve most of the explaining away deficiencies that occur in practice. The additional parameter α (which we call the 'explaining

- *N. E. Fenton and M. Neil are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, and Agena Ltd. E-mail: {n.fenton, m.neil}@qmul.ac.uk.*
- *T. Noguchi is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS. E-mail: t.noguchi@qmul.ac.uk.*

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2891680, IEEE Transactions on Knowledge and Data Engineering

away parameter') is used to make just a single adjustment to the CPT of the standard leaky noisy-OR – namely an adjustment to the probability of y being true when all parents are true. Finally, in Section 6 we conclude by describing the important and immediate practical benefits of the proposed extended leaky noisy-OR.

2 Related work

Before providing the necessary formal definition of noisy-OR and its extensions we briefly review its history and related work. The origins of noisy-OR are somewhat unclear. Pearl was the first to use the term [5], but he did so in a way that suggested it was already well-known. In a personal communication with the first author (while our paper was being prepared) Pearl said:

For me, as an engineer, there was no question that this function deserves the name "noisy-or", so I used it without checking if someone else used it before. If you find an earlier use, I will not be offended.

In fact, we were unable to find any previous use of the term, although something essentially equivalent to noisy-OR can be found (with difficulty) in the papers of Good [18], [19]. There is less doubt that the leaky noisy-OR was first proposed by Henrion [20].

The CII property of noisy-OR was identified in [15], and also in [16] where it is termed *reverse independence*. Researchers have sought – as we do here – to 'break' the property by extending noisy-OR. The solution in [17] - called *recursive noisy-OR* works but requires, in the worst case, all of the CPT parameters to be redefined. This is discussed fully in Section 4.

The noisy-OR has also been generalized to consider nonbinary (multistate) causes and effects [21]–[23]. These generalizations subsume the original noisy-OR function and do not resolve the CII issue More generally, multistate causes and effects can be modeled with the ranked node [3], [24] and the softmax function (one of the most widely used functions when parameter values need to be learned from data).

To learn parameters for the noisy-OR function, the expectation maximization algorithm is typically applied. When the network has high treewidth, the required expectations might be approximated using Monte Carlo or variational methods [25]. The extensions to these learning approaches have been proposed to avoid computation of expectations [11], [26].

3 Noisy-OR: Definition, benefits and limitations

In what follows we assume that we have a BN fragment in which a Boolean node y has n Boolean parents $x_1, \ldots, x_n$. In general, the CPT for y is a 2×2^n table requiring us to determine 2^n parameters, corresponding to probabilities that each of the parent state combinations is true. However, for some problems the known real-world relationship between the variables leads to the possibility of defining the CPT more concisely.

As an extreme example, suppose we are trying to model and predict failure in a system that is composed of n independent components, any one of which will cause the system to fail, and that the system cannot fail for any reason other than a component failure. If y represents the Boolean variable "System fails" and x_i represents Boolean variable "i^{th} component fails" then the CPT for y is defined simply by the logical OR function – we do not need to elicit individual probabilities for each parent state combination. However, in practice the assumptions required here for the OR function may be unrealistic. There may be uncertainty about whether a component failure will lead to a system failure and there may be other, unspecified, causes of system failure. Nevertheless, assuming still that the components fail independently, this is an example of a very common scenario in which the relationship between the child and its parents is 'similar' to the logical OR function. Another common situation of this type is in medical risk where, for example, 'smoking', 'lack of exercise', 'poor diet', 'stress' and 'family history of heart failure' are five potentially independent causes of a person suffering a heart attack before the age of 65, but none is certain to cause it. Also, even if all factors are false, it is still possible that a person may suffer a heart attack before the age of 65.

For such situations a noisy form of the OR function, called the ***noisy-OR*** function - and its extension the ***leaky noisy-OR*** - have been widely used as an approximation of the required real-world problem. Since the leaky noisy-OR is a more general and useful form we will focus on that only. This function requires only $n + 1$ parameters to be elicited, namely n weight parameters (one corresponding to each parent) and a leak factor. Formally:

Formal definition of the leaky noisy-OR function. Let $x_1, \ldots, x_n$ be n Boolean variables. For each $i = 1, \ldots, n$ let v_i be a number between 0 and 1 (we call v_i the **weight** associated with x_i) and let λ be a number between 0 and 1, which we call the **leak factor**. Let y be a Boolean variable with parents, $x_1, \ldots, x_n$. Writing 1 and 0 to represent Boolean states True and False respectively, we define the leaky noisy-OR function as:

$$P^{no}(y = 1 | x_1, \ldots, x_n) = 1 - (1 - \lambda) \prod_{i=1}^{n} (1 - v_i)^{x_i}$$

The original (standard) noisy-OR function is the leaky noisy-OR function when the leak factor $\lambda = 0$. If all the weight values are equal to 1 and the leak factor is 0, then the leaky noisy-OR function is equivalent to the Boolean OR function.

Since most of the literature – and all BN tools that implement it – refer to the leaky noisy-OR as simply the noisy-OR, for simplicity in what follows we do the same.

Just as if we were using the Boolean OR function, the noisy-OR requires us to assume that we can consider each of the $x_1, \ldots, x_n$ as 'causes' independently of the others in terms of their effect on y. However, this assumption makes sense for a large class of real world problems [6]–[10], and the properties of noisy-OR also make it easy to elicit the parameters from domain experts in practice in a way which makes them meaningful. For example, if $x_i = 0$ for all i then the probability $y = 1$ is simply equal to the leak factor (so to elicit the leak factor, we could ask the domain expert: "what is the probability that a patient who has none of the risk factors will still get the disease?"). Instead, if the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2891680, IEEE Transactions on Knowledge and Data Engineering

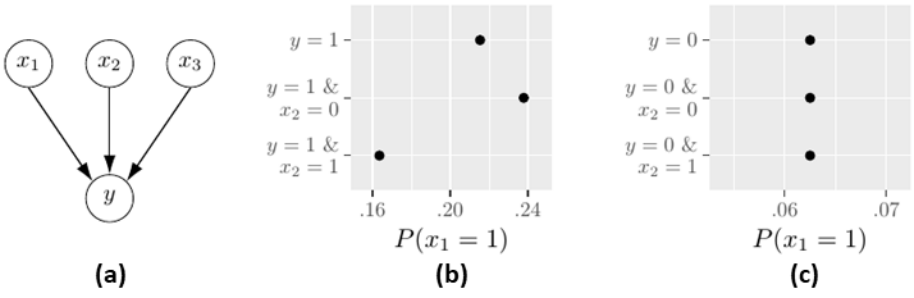

Fig. 1. The noisy-OR function of Example 1 (where $v_1 = 0.4$, $v_2 = 0.2$, $v_3 = 0.3$, and $\lambda = 0.1$). (a) Network structure (b) Shows that, given $y = 1$, $P(x_1)$ depends on the value of x_2. (c) Given $y = 0$, $P(x_1)$ is independent of x_2.

leak factor is 0 and $x_i = 1$ for exactly one i (with $x_j = 0$ for all $j \neq i$) then the probability that $y = 1$ is simply equal to the weight v_i associated with x_i (so we could elicit the weight values by asking the domain expert for each of the risk factors: "what is the probability that a person will have the disease if this risk factor is present and none of the others are?"

So, the noisy-OR is a very good approximation for the expected behaviour in these kinds of problems when it comes to forward inference (observations of causes on effect). It is also fine for backward inference when $y = 1$, since in this case the usual expected 'explaining away' behaviour is preserved. Specifically, for all $i \neq j$,

$$P^{no}(x_i = 1 | y = 1, x_j = 1) < P^{no}(x_i = 1 | y = 1).$$

In other words, if we observe that y is true and then also observe one parent x_j is true; then the fact that y is true is already explained by x_j being true – so the probability that each of the other x_i's is true decreases.

In addition, for all $i \neq j$,

$$P^{no}(x_i = 1 | y = 1, x_j = 0) > P^{no}(x_i = 1 | y = 1).$$

In other words, if we observe that y is true and then also observe one parent x_j is false; then the fact that y is true must have been caused by something else – so the probability that each of the other x_i's is true increases.

Henceforth we use the following which is archetypal of the kind of problem encountered in security and terrorism analysis [27].

Example 1. In any given week a terrorist organisation may or may not carry out an attack (y). There are several independent cells (x_1, …, x_n) in this organisation for which it may be possible in any week to determine heightened activity. If it is known that there is no heightened activity in any of the cells, then an attack is unlikely. However, for any cell if it is known there is heightened activity then there is a chance an attack will take place. The more cells known to have heightened activity the more likely an attack is.

An example formulation of this problem with noisy-OR is illustrated in Figure 1. The middle panel in Figure 1 shows the probability conditioned on $y = 1$. When conditioned on one of the cells having a heightened activity (i.e., $x_2 = 1$), the probability of the other cells having heightened activity decreases.

However, as we explain in the next section, the explaining away behaviour is ***not*** preserved when $y = 0$.

4 Causal Independence in the Noisy-OR Function and How to Break It

In Example 1, we saw that the noisy-OR model works well when an attack is observed (i.e., $y = 1$). Now we consider the case when we observe no attack (i.e., $y = 0$). In such situations, if it is known that there is heightened activity in one cell (e.g., $x_j = 1$), we expect the lack of attack to be explained away by a decrease in the probability that the remaining cells have heightened activity. In other words, in general we expect the following conditional dependence property to hold: for all $i \neq j$,

$$P(x_i = 1 | y = 0, x_j = 1) < P(x_i = 1 | y = 0).$$

This also extends to subsequent observations: letting $\mathcal{K}$ be a set of m integers between 1 and n ($0 \leq m \leq n - 2$), for all $i \neq j$, $i, j \notin \mathcal{K}$,

$$P(x_i = 1 | y = 0, x_k = 1 \; \forall \; k \in \mathcal{K}) > P(x_i = 1 | y = 0, x_j = 1, x_k = 1 \; \forall \; k \in \mathcal{K}) \quad (property\ 1)$$

We will refer to this desired behaviour as ***conditional dependence property 1***.

Less critical, but still important is what was expected if, having observed no attack (i.e. $y = 0$) we then discover that one of the cells had no heightened activity (say $x_j = 0$). In this case the absence of activity in this cell already partly explains why no attack happened and, hence, a small increase is expected (compared to before observing $x_j = 0$) in the probability that the remaining cells are active. In other words, in general we may expect the following conditional dependence property to hold: for all $i \neq j$,

$$P(x_i = 1 | y = 0, x_j = 0) > P(x_i = 1 | y = 0). \quad (property\ 2)$$

We will refer to this desired behaviour as ***conditional dependence property 2.***

Example 1 above is one of many real-world problems where experts and modellers had assumed the noisy-OR to be a good approximation and where it was assumed that (1) would hold as a matter a course and that (2) was desirable.

Unfortunately, as proved in [15], noisy-OR has the the CII property. What this means is that given $y = 0$, x_i and x_j are independent for all $i \neq j$ (see the right panel in Figure 1). Thus, neither (1) nor (2) holds for the noisy-OR. This CII stems from the causal independence assumption that defines the noisy-OR: the causal influence of a cause on an effect is independent of the other causes. Under this assumption, the absence of the effect indicates that each cause independently failed to cause the effect.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2891680, IEEE Transactions on Knowledge and Data Engineering

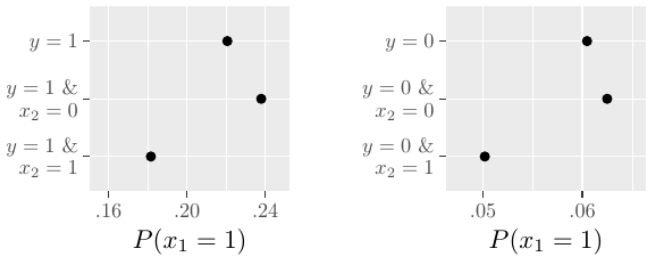

Fig. 2. Extension of noisy-OR applied to Example 1 (Fig. 1), where parameter $\alpha = 1 - P(y = 0|x_1 = x_2 = x_3 = 1) = 0.9$, replaces the value $1 - P(y = 0|x_1 = x_2 = x_3 = 1) = 0.6976$. In contrast to Fig 1, $P(x_1)$ depends on the value of x_2 regardless of the value of y.

We can therefore break the CII by breaking the causal independence assumption. The causal dependence takes one of the three forms: *synergy*, *interference*, or *inhibition* [17]. To satisfy (1), however, only synergy is required. Synergy occurs where the combination of causes has a greater influence than the combined independent product. This synergy also partly satisfies (2).

To model synergy, the ***recursive noisy-OR*** function has been proposed [17]. This requires experts' inputs on joint causal influences as well as on singleton causal influences. For example when two causes, x_1 and x_2, are considered, the expert provides three values: the probability of y equalling 1 when both x_1 and x_2 equal 1; the probability of y equalling 1 when only x_1 equals 1; and the probability of y equalling 1 when only x_2 equals 1.

Unfortunately, the number of required inputs for the recursive noisy-OR increases exponentially with the number of causes (although missing inputs can be estimated when they are not provided). Hence, while the recursive noisy-OR is complete in the sense of allowing the synergy and the interference of causes, it is not a practically useful solution because it introduces elicitation complexities that essentially defeat the objective of using a noisy-OR. What we do in the next section is define a new special case of the recursive noisy-OR that requires only a single additional parameter to the noisy-OR. We prove it is sufficient to satisfy property (1) and partially satisfy property (2).

5 AN EXTENSION TO THE NOISY-OR

Here we provide an extension to the noisy-OR and prove that it satisfies conditional dependence property (1), as illustrated in Figure 2. In addition, the extension satisfies conditional dependence property (2) in the case where exactly one (but no more) of the x_i is observed.

The extension involves introducing a new 'explaining-away' parameter, α $(0 < \alpha < 1)$. Formally, $\alpha = 1 - P(y = 0|x_1 = \cdots = x_n = 1)$, where the conditional probability $P(y = 0|x_1 = \cdots = x_n = 1)$ is predefined and:

$$P^{ext}(y = 1|x_1, \ldots, x_n) = \begin{cases} 1 - (1-\alpha)(1-\lambda)\prod_{i=1}^{n}(1-v_i)^{x_i} & if\ x_1 = \cdots = x_n = 1 \\ 1 - (1-\lambda)\prod_{i=1}^{n}(1-v_i)^{x_i} & otherwise \end{cases}$$

Therefore, this extension differs from the noisy-OR function only when $x_1 = \cdots = x_n = 1$. An example application is illustrated in Fig.2. Section 6 explains how the proposed extension is trivially implemented when using standard BN software tools.

5.1 Proof that the extension satisfies property (1)

In this section, we show that the extension satisfies property (1). Letting $\mathcal{K}$ be a set of m integers between 1 and n $(0 \le m \le n-2)$, for all $i \ne j$, $i, j \notin \mathcal{K}$, we show that

$$P^{ext}(x_i = 1|y = 0, x_k = 1\ \forall\ k\ \in\ \mathcal{K}) > P^{ext}\big(x_i = 1\big|y = 0, x_j = 1, x_k = 1\ \forall\ k\ \in\ \mathcal{K}\big).$$

To begin, we simplify the notations by denoting $p_i = P(x_i = 1)$, $\psi = P^{no}(y = 0|x_i = 1, x_j = 1, x_k = 1\ \forall\ k\ \in\ \mathcal{K})$, and $\phi = P^{no}(y = 0|x_1 = \cdots = x_n = 1)$, and $\gamma = \prod_{l\notin\mathcal{K}, l\ne i, l\ne j} p_l$.

Then, by the definition of the extension,

$$P^{\text{ext}}(y = 0 \mid x_i = 1,\ x_j = 1,\ x_k = 1\ \forall\ k \in \mathcal{K}) = \psi - \gamma\phi + \gamma(1-\alpha)\phi = \psi - \alpha\gamma\phi,$$

$$P^{\text{ext}}(y = 0 \mid x_i = 1,\ x_j = 0,\ x_k = 1\ \forall\ k \in \mathcal{K}) = \psi/(1 - v_j), \text{and}$$

$$P^{\text{ext}}(y = 0 \mid x_i = 0,\ x_j = 0,\ x_k = 1\ \forall\ k \in \mathcal{K}) = \psi/\big[(1 - v_i)(1 - v_j)\big].$$

To evaluate the left hand-side of (1), we marginalize out x_j from the conditioning:

$$P^{ext}(y = 0|x_i = 1, x_k = 1\ \forall\ k\ \in\ \mathcal{K}) = p_j P^{ext}\big(y = 0\big|x_i = 1, x_j = 1, x_k = 1\ \forall\ k\ \in\ \mathcal{K}\big) + \big(1 - p_j\big)P^{ext}\big(y = 0\big|x_i = 1, x_j = 0, x_k = 1\ \forall\ k\ \in\ \mathcal{K}\big) = \big[p_j(1 - v_j)(\psi - \gamma\phi\alpha) + \psi(1 - p_j)\big]/(1 - v_j)$$

A similar calculation gives us $\mathrm{P}^{\text{ext}}(\mathrm{y} = 0 \mid \mathrm{x_j} = 1, \mathrm{x_k} = 1\ \forall\ \mathrm{k} \in \mathcal{K})$ and $\mathrm{P}^{\text{ext}}(\mathrm{y} = 0 \mid \mathrm{x_k} = 1\ \forall\ \mathrm{k} \in \mathcal{K})$. Then, it is straightforward to derive

$$\frac{P^{\text{ext}}(y = 0 \mid x_k = 1\ \forall\ k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 1,\ x_k = 1\ \forall\ k \in \mathcal{K})\, p_i} + \xi_1 = \frac{P^{\text{ext}}(y = 0 \mid x_j = 1,\ x_k = 1\ \forall\ k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 1,\ x_j = 1,\ x_k = 1\ \forall\ k \in \mathcal{K})\, p_i}$$

where

$$\xi_1 = \frac{\alpha\gamma\phi\psi(1 - p_i)(1 - p_j)}{(1 - v_i)(\psi - \alpha\gamma\phi)[\alpha\gamma\phi(v_j p_j - p_j) + \psi(1 - v_j p_j)]}$$

Here ξ_1 is a nonnegative term, because all the terms in its numerator and denominator are nonnegative. Below, we prove the second and third terms in the denominator, as the other terms are all nonnegative by their definitions. The second term in the denominator is

$$\psi - \alpha\gamma\phi = P^{\text{ext}}(y = 0 \mid x_i = 1,\ x_j = 1,\ x_k = 1\ \forall\ k \in \mathcal{K}) \ge 0$$

Given the immediate above, $\psi \ge \alpha\gamma\phi$, and thus, the third term in the denominator is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2891680, IEEE Transactions on Knowledge and Data Engineering

$$\begin{aligned}&\alpha\gamma\phi(v_j p_j - p_j) + (1 - v_j p_j)\psi\\ &\geq \alpha\gamma\phi(v_j p_j - p_j) + \alpha\gamma\phi(1 - v_j p_j)\\ &= \alpha\gamma\phi(1 - p_j) \geq 0.\end{aligned}$$

Therefore,

$$\begin{aligned}&\xi_1 \geq 0\\ &\Leftrightarrow P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_k = 1\,\forall\, k \in \mathcal{K})^{-1}\\ &\quad \leq P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_j = 1,\, x_k = 1\,\forall\, k \in \mathcal{K})^{-1}\\ &\Leftrightarrow P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_k = 1\,\forall\, k \in \mathcal{K})\\ &\quad \geq P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_j = 1,\, x_k = 1\,\forall\, k \in \mathcal{K}).\end{aligned}$$

5.2 Proof that the extension satisfies property (2)

Now we show (2) holds only when $m = 0$ and $\mathcal{K}$ is an empty set: for all $i \neq j$ and $i, j \notin \mathcal{K}$, we show

$$\begin{cases} P^{ext}(x_i = 1 | y = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}) \\ = P^{ext}\big(x_i = 1 \big| y = 0, x_j = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big) & if\ m > 0 \\ P^{ext}(x_i = 1 | y = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}) \\ < P^{ext}\big(x_i = 1 \big| y = 0, x_j = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big) & if\ m = 0 \end{cases}$$

We introduce a new notation:

$$\omega \ = P^{\text{no}}(y = 0 \mid x_i = 0, x_j = 0, x_k = 0\,\forall\, k \in \mathcal{K}).$$

Then,

$$P^{ext}\big(y = 0 \big| x_i = 1, x_j = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big) = \omega(1 - v_i),$$

and

$$P^{ext}\big(y = 0 \big| x_i = 0, x_j = 1, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big) = \omega\big(1 - v_j\big).$$

Marginalizing out x_j from the conditioning, we obtain

$$\begin{aligned}&P^{ext}(y = 0 | x_i = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K})\\ &= p_j P^{ext}\big(y = 0 \big| x_i = 0, x_j = 1, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big)\\ &+ \big(1 - p_j\big)P^{ext}\big(y = 0 \big| x_i = 0, x_j = 0, x_k = 0\ \forall\ k\ \in\ \mathcal{K}\big)\\ &= \omega\big(1 - p_j\big) + \omega\big(1 - v_j\big)p_j = \omega\big(1 - p_j v_j\big),\end{aligned}$$

and hence,

$$\begin{aligned}&P^{\text{ext}}(y = 0 \mid x_i = 1,\, x_k = 0\,\forall\, k \in \mathcal{K})\\ &= \begin{cases} P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})(1 - v_i) & \text{if } m > 0 \\ P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})(1 - v_i) & \\ \quad + P^{\text{ext}}(y = 0 \mid x_1 = x_2 = \cdots = x_n = 1)\displaystyle\prod_{l \neq i} p_l & \text{if } m = 0 \end{cases}\\ &= \begin{cases} \omega(1 - p_j v_j)(1 - v_i) & \text{if } m > 0 \\ \omega(1 - p_j v_j)(1 - v_i) - \alpha\gamma\phi & \text{if } m = 0. \end{cases}\end{aligned}$$

Then,

$$\begin{aligned}&P^{\text{ext}}(y = 0 \mid x_k = 0\,\forall\, k \in \mathcal{K})\\ &= p_i P^{\text{ext}}(y = 0 \mid x_i = 1,\, x_k = 0\,\forall\, k \in \mathcal{K})\\ &\quad + (1 - p_i)P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\\ &= \begin{cases} \omega(1 - p_i v_i)(1 - p_j v_j) & \text{if } m > 0 \\ \omega(1 - p_i v_i)(1 - p_j v_j) - \alpha\gamma\phi p_i & \text{if } m = 0 \end{cases}\end{aligned}$$

A straightforward arithmetic shows that

$$\begin{aligned}&\frac{P^{\text{ext}}(y = 0 \mid x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i} + \xi_2\\ &= \frac{P^{\text{ext}}(y = 0 \mid x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i}\end{aligned}$$

where $\xi_2 \geq 0$. Specifically,

$$\xi_2 = \begin{cases} 0 & \text{if } m > 0 \\ \alpha\gamma\phi p_i / \big[\omega(1 - v_j p_j)\big] & \text{if } m = 0 \end{cases}$$

Therefore, when $m > 0$, $\xi_2 = 0$

$$\begin{aligned}&\Leftrightarrow \frac{P^{\text{ext}}(y = 0 \mid x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i}\\ &= \frac{P^{\text{ext}}(y = 0 \mid x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i}\\ &\Leftrightarrow P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\\ &= P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K}).\end{aligned}$$

Instead, if $m = 0$, $\xi_2 > 0$

$$\begin{aligned}&\Leftrightarrow \frac{P^{\text{ext}}(y = 0 \mid x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i}\\ &< \frac{P^{\text{ext}}(y = 0 \mid x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})}{P^{\text{ext}}(y = 0 \mid x_i = 0,\, x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\, p_i}\\ &\Leftrightarrow P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_k = 0\,\forall\, k \in \mathcal{K})\\ &< P^{\text{ext}}(x_i = 1 \mid y = 0,\, x_j = 0,\, x_k = 0\,\forall\, k \in \mathcal{K}).\end{aligned}$$

5.3 Suppressing undesired behavior

Our extension implements the synergy of all the causes: their combination has a greater influence on the effect than the combined independent product (the Noisy-OR). The strength of this synergy is determined by the parameter α. As we discussed above, this synergy enables the explaining-away behaviour when $y = 0$. When the synergy is very strong, however, the synergy may produce undesired behaviour when $y = 1$. In particular, when $y = 1$, conditioning on $x_1 = 1$ can increase the probability that $x_2 = 1$: once we know one cause was present/true, probability that another cause was present/true increases. This behaviour is undesired, because it is inconsistent with the explaining-away behaviour we expect. To suppress this undesired behaviour, α needs to be less than a certain value to satisfy the following:

$$\begin{aligned}&P(x_i = 1 | y = 1, x_j = 1, x_k = 1 \forall k \in \mathcal{K})\\ &< P(x_i = 1 | y = 1, x_k = 1 \forall k \in \mathcal{K})\\ &\Leftrightarrow \frac{P(y = 1 | x_i = 1, x_j = 1, x_k = 1 \forall k \in \mathcal{K})\, p_i}{P(y = 1 | x_j = 1, x_k = 1 \forall k \in \mathcal{K})}\\ &< \frac{P(y = 1 | x_i = 1, x_k = 1 \forall k \in \mathcal{K})\, p_i}{P(y = 1 | x_k = 1 \forall k \in \mathcal{K})}.\end{aligned}$$

This inequality is satisfied when

$$\alpha < \frac{v_i v_j \psi}{\gamma\phi\big[(1 - v_i)(1 - v_j) - \psi\big]}$$

for any i and j. This upper bound is larger when $p(x_l = 1)$ is smaller for an unobserved cause x_l or when v_i is larger for an observed cause x_i.

6 Conclusions

The leaky noisy-OR is an efficient method in practice to approximate the required CPT of an effect node with multiple 'independent causal' parents. However, because of the CII property, the approximation fails to capture all 'explaining away' behaviour in situations where it is commonly used (such as in our example of security and intelligence analysis). In other words, it may be providing results that do not properly match known empirical observations for the problem being modelled. We have described a simple extension to leaky noisy-OR that requires domain experts to provide only one extra parameter in order to capture most of the conditional dependence properties that are required in such practical situations. Our solution contrasts with previous solutions that, in the worst case, require an exponential number of extra parameters to be elicited. Since software tools already automatically generate a full CPT from a leaky noisy-OR expression, it is trivial to use the extension proposed in

this paper: as a result of our proof, all that is needed is to adjust (by the 'explaining away' parameter) the CPT entry corresponding to the case where all parent nodes are true. This solution is now being used in a number of onging projects aimed at improving decision-making.

ACKNOWLEDGMENT

We acknowledge support by: the European Research Council under project, ERC-2013-AdG339182 (BAYES_KNOWLEDGE); the Leverhulme Trust under Grant RPG-2016-118 CAUSAL-DYNAMICS; and Agena Ltd for software support. We also acknowledge Nicole Cruz for identifying the need to constrain the size of the additional parameter (Section 5.3) and the helpful recommendations and comments of Judea Pearl.

REFERENCES

[1] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artif. Intell.*, vol. 29, no. 3, pp. 241–288, 1986.

[2] M. K. Druzdzel and L. C. Van der Gaag, "Elicitation of probabilities for belief networks: combining qualitative and quantitiative information," *Proc 11th Ann Conf on Uncertainty in Artifical Intelligence (UAI-95), 141-148, Montreal, Quebec, Canada, August*. 1995.

[3] N. E. Fenton, M. Neil, and J. Gallan, "Using Ranked nodes to model qualitative judgements in Bayesian Networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1420–1432, 2007.

[4] S. Monti and G. Carenini, "Dealing with the Expert Inconsistency in Probability Elicitation," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 4, pp. 499–508, 2000.

[5] J. Pearl, *Probabilistic reasoning in intelligent systems*. Palo Alto, CA: Morgan Kaufmann , 1988.

[6] V. Anand and S. M. Downs, "Probabilistic asthma case finding: a noisy or reformulation.," in *Annual Symposium proceedings. AMIA Symposium*, 2008, vol. 2008, pp. 6–10.

[7] M. Pradhan, G. M. Provan, B. Middleton, and M. Henrion, "Knowledge Engineering for Large Belief Networks," in *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI1994)*, 1994.

[8] S. P. D. Woudenberg and L. C. van der Gaag, "Using the Noisy-OR Model Can Be Harmful … But It Often Is Not," in *ECSQARU 2011: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Lecture Notes in Computer Science, vol 6717*, 2011, pp. 122–133.

[9] A. Oniśko, M. J. Druzdzel, and H. Wasyluk, "Learning Bayesian network parameters from small data sets: application of Noisy-OR gates," *Int. J. Approx. Reason.*, vol. 27, no. 2, pp. 165–182, Aug. 2001.

[10] J. H. Bolt and L. C. van der Gaag, "An Empirical Study of the Use of the Noisy-Or Model in a Real-Life Bayesian Network," Springer, Berlin, Heidelberg, 2010, pp. 11–20.

[11] Yoni Halpern and D. Sontag, "Unsupervised Learning of Noisy-Or Bayesian Networks," in *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

[12] T. L. Griffiths and J. B. Tenenbaum, "Theory-based causal induction.," *Psychol. Rev.*, vol. 116, no. 4, pp. 661–716, Oct. 2009.

[13] Agena Ltd, "AgenaRisk." http://www.agenarisk.com, 2018.

[14] Hugin A/S, "Hugin Expert," 2018. [Online]. Available: http://www.hugin.com.

[15] J. M. Agosta, "Conditional inter-causally independent node distributions, a property of noisy-or models," in *In Proceedings of the seventh conference on uncertainty in artificial intelligence,* 1991, pp. 9–16.

[16] F. G. Cozman, "Axiomatizing noisy-or," in *Proceedings of the 16th European conference on Artificial Intelligence*, 2004, pp. 981–982.

[17] J. F. Lemmer and D. E. Gossink, "Recursive Noisy OR—A Rule for Estimating Complex Probabilistic Interactions," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 6, pp. 2252–2261, Dec. 2004.

[18] I. J. Good, "A Causal Calculus (I)," *The British Journal for the Philosophy of Science*, vol. 11. Oxford University PressThe British Society for the Philosophy of Science, pp. 305–318, 1961.

[19] I. J. Good, "A Causal Calculus (II)," *The British Journal for the Philosophy of Science*, vol. 12. Oxford University PressThe British Society for the Philosophy of Science, pp. 43–51, 1961.

[20] M. Henrion, "Some Practical Issues in Constructing Belief Networks," *Uncertainty in Artificial Intelligence 3*. North Holland: Elsevier Science, pp. 161–173, 1989.

[21] F. J. Díez, "Parameter adjustment in Bayes networks: the generalized noisy or-gate," *Ninth Conference on Uncertainty in Artificial Intelligence*. Washington D.C, pp. 99–105, 1993.

[22] S. Srinivas, "A Generalization of the Noisy-Or Model," *Ninth Conference on Uncertainty in Artificial Intelligence*. pp. 208–218, 1993.

[23] J. Vomlel, "Generalizations of the Noisy-OR model." [Online]. Available: http://staff.utia.cas.cz/vomlel/Voml_3484.pdf. [Accessed: 11-Oct-2017].

[24] P. Laitila and K. Virtanen, "Improving Construction of Conditional Probability Tables for Ranked Nodes in Bayesian Networks," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2016.

[25] T. Singliar and M. Hauskrecht, "Noisy-or component analysis and its application to link analysis," *J. Mach. Learn. Res.*, vol. 7, pp. 2189–2213, 2006.

[26] Y. Jernite, Y. Halpern, and D. Sontag, "Discovering Hidden Variables in Noisy-Or Networks using Quartet Tests," *Neural Inf. Process. Syst.*, vol. 26, pp. 2355–2363, 2013.

[27] Bayesian Argumentation via Delphi. IARPA funded project. https://bayesiandelphi.org/