Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

A Noisy-OR Model for Continuous Time Bayesian Networks

Logan Perreault, Shane Strasser, Monica Thornton, and **John W. Sheppard**
Computer Science Department
Montana State University
Bozeman, MT 59717
{logan.perreault, shane.strasser, monica.thornton, john.sheppard}@montana.edu

Abstract

A continuous time Bayesian network is a graphical model capable of describing discrete state systems that evolve in continuous time. Unfortunately, the number of parameters required for each node in the graph is exponential in the number of parents of the node, which can be prohibitively large for many real-world systems. To mitigate this problem, we propose a Noisy-OR model for continuous time Bayesian networks, which can reduce the number of required parameters from exponential to linear. We describe the model, as well as the process required to compute the remaining unspecified parameters. Finally, we experimentally validate the correctness of the proposed Noisy-OR formulation.

Introduction

Many-real world problems can be solved by modeling the state of a system as it evolves over time. If the states of that system are discrete, continuous time Markov processes (CTMPs) provide an effective framework for achieving this task. Unfortunately, the size of a CTMP is exponential in the number of variables in the system. To mitigate this problem, continuous time Bayesian networks (CTBNs) have been proposed as a model that factors the underlying CTMP using conditional independencies (Nodelman, Shelton, and Koller 2002). These conditional independencies are encoded in a directed graph structure so that each node need only describe the state transition behavior for a single variable given each instantiation of the parent states. The size of the resulting model is exponential in the largest parent set in the graph.

Although this framework significantly reduces the number of necessary parameters, for some applications the model may still be too complex. If even a single node in the network has a large parent set, the complexity of the model may make learning and inference tasks intractable. Furthermore, the space required to store the model in memory may be unreasonable, resulting in slow query response times. To avoid this, parameterizing the model must be simplified further.

This problem has been handled in the Bayesian network (BN) community using numerous methods, one of the most popular being the Noisy-OR model. Noisy-OR works by assuming there is a disjunctive interaction among the parents of a node, rather than a conjunctive interaction. This assumption is usually interpreted in the context of a cause and effect model where each parent node is seen as an event sufficient to cause behavior in the child node. For this reason, the Noisy-OR model requires only that a node be parameterized for the instances where a single parent event occurs, rather than every parent-state combination. The resulting model is linear in the size of the largest parent set rather than exponential, as is usually the case with BNs.

Although the Noisy-OR model and several extensions have been studied thoroughly in the context of BNs, no such research has been presented in the CTBN literature. This is a significant limitation for CTBNs, especially when dealing with real-world networks that exhibit disjunctive causal relationships with a large number of causes. In this paper, we present the Noisy-OR model as it works in continuous time and describe how it can be used with CTBNs.

Background

To describe how Noisy-OR can be extended for CTBNs, we begin by providing a brief overview of how CTBNs operate and how Noisy-OR works in the context of BNs. For more information about CTBNs, the reader is referred to Nodelman's original work on the subject (Nodelman, Shelton, and Koller 2002). For additional material on the Noisy-OR model, see the work of Pearl and other authors listed in the Noisy-OR background section (Pearl 1988).

BNs provide a factored representation of a joint probability distribution over a set of variables. This is achieved by using a directed graph to encode conditional independencies in the distribution. The nodes in the graph represent the variables in the joint distribution, while the edges describe direct conditional dependencies. Each node is parameterized using a conditional probability table (CPT) that indicates the probability of the variable being in each state conditioned on each instantiation of the states of the parents in the network.

While BNs provide compact representations capable of modeling static systems, some problems require reasoning about systems that evolve in continuous time. A CTMP consists of an initial distribution P over the set of states in a system, and an intensity matrix Q that describes the transition behavior between states over time. More specifically, the entry q_{ij} in row i, column j of matrix Q indicates that the time it takes to transition from state i to state j is drawn from an exponential distribution with rate q_{ij}. Diagonal entries q_{ii} are constrained to be the negative sum of the remaining entries in the row, such that each row sums to zero. The entire time that the system spends in a state i before transitioning to another state is characterized by an exponential distribution with a rate of $-q_{ii}$. Since the transition is drawn from an exponential distribution, the expected time that the system spends in state i is $1/q_{ii}$.

Continuous time Bayesian networks (CTBNs) borrow many concepts from Bayesian networks, but rather than factoring a joint probability distribution, a CTBN factors a CTMP. The initial distribution and intensity matrix for a CTMP are exponential in the size of the variables for a system, since each row or column represents a full instantiation to every variable. To avoid this problem, a CTBN uses a directed network structure much like the one used in a BN to encode conditional independencies. This allows the full intensity matrix for the Markov process to be decomposed into a set of conditional intensity matrices (CIMs) for each node. One CIM is defined for each instantiation of a node's parents, and each CIM describes the state transition behavior over a state-space local to the node's variable.

Related Work

Pearl first proposed a binary Noisy-OR model for BNs, and in this work he argued that a system qualifies for Noisy-OR if it meets the assumptions of accountability and exception independence (Pearl 1988). Accountability refers to the notion that at least a single parent must be *True* if a child is observed to be *True*, since the model can be interpreted as causal. Exception independence means that the factors that inhibit causation are independent of one another. This can be interpreted as a series of AND gates for each parent, where the inputs include the cause and a negated inhibitor mechanism. The output of these AND gates are then passed through as inputs to an OR gate, which produces a Noisy-OR model. Finally, Pearl shows how parameters in the CPTs can be calculated on the fly using this model.

The Noisy-OR gate has been applied successfully to a variety of real-world problems that require modeling disjunctive interaction in BNs. Oniśko *et al.* use Noisy-OR to reduce the number of parameters that must be specified for use in small datasets (Oniśko, Druzdzel, and Wasyluk 2001). The authors learn the reduced set of parameters in the BN by using a small set of patient records for the purpose of diagnosing liver disorders. Murphy and Mian review methods for learning dynamic BNs for the purpose of modeling gene expression data, and describe how Noisy-OR gates allow for compact parameterization of the CPTs in the networks (Murphy and Mian 1999). Bobbio *et al.* show how to convert fault trees into BNs by using Noisy-OR nodes, which capture the disjunctive causation assumed by a fault tree (Bobbio et al. 2001). Strasser and Sheppard provide an automated method for deriving a BN from a D-matrix, which is a model that describes the relationships between system faults and the tests that monitor them (Strasser and Sheppard 2013). They specify the test nodes to use the Noisy-OR gate model, which retains the original semantics of the fault-test relationships described by the D-matrix.

There have been several extensions to the Noisy-OR model since its original formulation. Henrion extends the original Noisy-OR model by relaxing the assumption that a variable may only enter the *True* state if a parent is in the *True* state (Henrion 1987). This is done by introducing another parent called a "leak" node that is always on, which represents the potential for unmodeled events to cause the child to become *True*. Much of our work is based on research performed by Srinivas, which further generalizes the Noisy-OR model (Srinivas 1993). While Pearl's original model assumes binary variables and a logical OR function, Srinivas' generalization allows for multi-state variables and the use of any discrete function. In concurrent research, Diez also describes a multi-state version of the Noisy-OR model (Diez 1993). Finally, Heckerman discusses the Noisy-OR model within the context of a more general framework called causal independence, which describes the notion of any independence that can be used to simplify the exponential parameterization required for the CPTs in a BN (Heckerman 1993; Heckerman and Breese 1994; 1996).

The Noisy-OR model has been studied within the context of BNs, but no work has been done to extend its usage to CTBNs. Cao demonstrated how CIMs may be parameterized to emulate AND and OR gates for the purpose of modeling fault trees within the CTBN framework (Cao 2011). This parameterization is accomplished by using infinity for the rates of the intensity matrix to ensure instantaneous transitions to a *True* state based on valid AND/OR configurations of the parents. Zero values are used to guarantee that a transition does not occur until a proper configuration is observed, and also that no transition occurs after moving into a *True* state. Unlike our approach, Cao is modeling a logical OR gate using CTBN nodes with deterministic transitions.

Survival analysis models describe systems using a survival function $S(t) = P(T > t)$, and the related lifetime distribution function $F(t) = P(T \leq t) = 1 - S(t)$ (Johnson 1999). These functions indicate the probability that an event occurs at some time t after or before a specified time of interest T. The derivatives of these functions are typically denoted as $s(t)$ and $f(t)$ respectively. When viewed as a causal network where nodes can transition from non-failing states to failing states, a CTBN can be thought of as a survival model where the lifetime distribution function is equal to the exponential distribution $F(t) = 1 - \exp(-\lambda t)$. This can be related back to the survival function as follows: $S(t) = 1 - (1 - \exp(-\lambda t)) = \exp(-\lambda t)$.

The hazard function for survival models is defined as the instantaneous event probability over time. A variety of parametric distributions are used in the survival analysis literature (Kleinbaum and Klein 2012). When using an exponential distribution, the hazard function is defined as

$$H(t) = \frac{f(t)}{S(t)} = \frac{\lambda \exp(-\lambda t)}{\exp(-\lambda t)} = \lambda,$$

which is the constant transition rate for the exponential distribution. While survival models are typically represented using a regression model with time as the sole response vari-

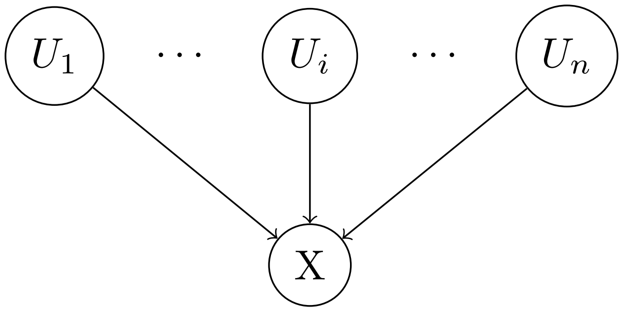

Figure 1: The Noisy-OR network structure.

able, a CTBN models the state of multiple variables that change as time progresses. Another notable difference is that CTBNs assume an exponential distribution, while survival models are less restrictive in terms of the supported distributions. There are many systems whose survival follow non-exponential distributions, and CTBNs have been extended to include more complex parametric distributions (Perreault et al. 2015).

Noisy-OR in Bayesian Networks

The Noisy-OR model is a generalized version of the logical OR gate, capable of capturing the non-determinism in a system with disjunctive interactions. There are n binary inputs $u_1, \dots, u_n$ and a single binary output x. The domain for each u_i and x consists of the states *False* and *True*, which we denote using 0 and 1 respectively. The output x takes on a value of 1 if one of the inputs u_i is in state 1.

The Noisy-OR model can be described generally as an OR gate with inputs $u_1, \dots, u_n$ and an output of x. The BN structure is shown in Figure 1, where the inputs are represented using nodes $U_1, \dots, U_n$, and the output is the node X. A lowercase u_i is used to refer to an instantiation of node U_i, and x is considered to be an instantiation of X. Without loss of generality, let $\tilde{\mathbf{u}}_i$ denote the specific set of parent node instantiations $\{u_0^0, \cdots, u_{i-1}^0, u_i^1, u_{i+1}^0, \cdots, u_n^0\}$, such that only node U_i has a value of 1, while the remaining variables in $\mathbf{U}$ are set to 0.

We say that if a single parent U_i is in state 1, it will cause X to be 1 with probability λ_i, and 0 with probability $(1-\lambda_i)$, as shown in the following:

$$P(x^1|\tilde{\mathbf{u}}_i) = \lambda_i$$
$$P(x^0|\tilde{\mathbf{u}}_i) = (1-\lambda_i).$$

The probability that X is in state 0 given its entire parent set $\mathbf{U}$ is calculated as the product of the probabilities $P(X = 0|\tilde{\mathbf{u}}_i)$ for each U_i that is in state 0. We denote the subset of parents that are in state 1 as $\tilde{\mathbf{u}}^+$, as shown below:

$$\begin{aligned} P(x^0|\mathbf{U}) &= \prod_{u_i \in \tilde{\mathbf{u}}^+} P(x^0|\tilde{\mathbf{u}}_i) \\ &= \prod_{u_i \in \tilde{\mathbf{u}}^+} (1-\lambda_i). \end{aligned}$$

$\mathbf{U}$	$\triangleq$	Set of parent variables: $\{U_1, \cdots, U_n\}$
$\mathbf{u}$	$\triangleq$	An instantiations of $\mathbf{U}$: $\{u_1, \cdots, u_n\}$
$\tilde{\mathbf{u}}_i$	$\triangleq$	The instantiation $\mathbf{u}$ where only $u_i = 1$: $\{u_0^0, \cdots, u_{i-1}^0, u_i^1, u_{i+1}^0, \cdots, u_n^0\}$
$\tilde{\mathbf{u}}^+$	$\triangleq$	The subset of parents in state 1.
$x^s[t]$	$\triangleq$	X is in state s at time t.
$x^s[t_1, t_2)$	$\triangleq$	X is in state s over the interval $[t_1, t_2)$.
$\phi_{\mathbf{u}}^0(t)$	$\triangleq$	Probability of not transitioning to state 1 after by time t given parent values $\mathbf{u}$: $P(x^0[t_1, t_2)|x^0[t_1])$
$\phi_{\mathbf{u}}^{0\to1}(t)$	$\triangleq$	Probability of transitioning to state 1 by time t given parent values $\mathbf{u}$: $P(x^0[t_1, t_1+\delta) \wedge x^1[t_1+\delta, t_2)|x^0[t_1])$

Table 1: Notation

We can use this to calculate the probability that X is in state 1 by taking 1 minus this probability, as follows:

$$\begin{aligned} P(x^1|\mathbf{U}) &= 1 - P(x^0|\mathbf{U}) \\ &= 1 - \prod_{u_i \in \tilde{\mathbf{u}}^+} (1-\lambda_i). \end{aligned}$$

Noisy-OR in Continuous Time Bayesian Networks

To describe Noisy-OR in CTBNs, we again use $\tilde{\mathbf{u}}_i$ to indicate an instantiation of the parent set $\mathbf{U}$ where only node U_i is in state 1. Let $[t_1, t_2)$ be a time interval from t_1 to t_2, such that the length of the interval is $t = t_2 - t_1$. Next, we use $x^0[t_1, t_2)$ to indicate that variable X is in state 0 during the interval $[t_1, t_2)$, and $x^1[t_1, t_2)$ to indicate that it is in state 1 during that interval. Similarly, we use $x^0[t]$ and $x^1[t]$ to indicate that the variable X is in state 0 or 1 at a discrete time t. Let $\phi^0(t)$ denote $P(x^0[t_1, t_2)|x^0[t_1])$. This is the probability that X remains in state 0 for the entire time interval $[t_1, t_2)$, and does not transition to 1. Next, let $\phi^{0\to1}(t)$ denote $P(x^0[t_1, t_1+\delta) \wedge x^1[t_1+\delta, t_2)|x^0[t_1])$, where δ is some discrete point in the time interval $[t_1, t_2)$. This is the probability that X transitions from state 0 to state 1 sometime during the time interval $[t_1, t_2)$. A subscript is used to show conditional dependence on a state instantiation of $\mathbf{U}$. A summary of the notation used throughout the following section is provided in Table 1.

The CIMs for node X are as follows:

$$Q_{X|\mathbf{u}} = \begin{matrix} & 0 & 1 \\ 0 & -\lambda_{\mathbf{u}} & \lambda_{\mathbf{u}} \\ 1 & \mu_{\mathbf{u}} & -\mu_{\mathbf{u}} \end{matrix}.$$

The parameters in these matrices describe the rate at which the variable is expected to transition to another state. Using this, the probability density function f and the cumulative distribution function F can be defined for transitioning from state 0 to state 1, as follows:

$$f_{\lambda_{\mathbf{u}}}(t) = \lambda \exp(-\lambda_{\mathbf{u}} t)$$
$$F_{\lambda_{\mathbf{u}}}(t) = 1 - \exp(-\lambda_{\mathbf{u}} t).$$

Assuming we are starting at time t_1, $F_{\lambda_{\mathbf{u}}}(t)$ defines the probability of transitioning from a state of 0 to a state of 1 by time $t_1 + t = t_2$. The probability of having transitioned from state 0 to state 1 by time t_2 is therefore given by $\phi_{\mathbf{u}}^{0\to 1}(t) = 1 - \exp(-\lambda_{\mathbf{u}} t)$. This means that $\phi_{\mathbf{u}}^{0}(t) = \exp(-\lambda_{\mathbf{u}} t)$.

The Noisy-OR network structure for the CTBN is the same as the BN shown in Figure 1. As with a BN, we assume that the presence of each parent on its own is capable of defining the behavior of X. As before, the goal is to parameterize X only for the cases where at most a single parent is in state 1. First, we assign initial distributions such that all nodes start in state 0 deterministically. Next, we parameterize the intensity matrix for the case where no parents are in state 1 to ensure that there is a 0 rate of transition to state 1. Finally, we parameterize each of the n CIMs $Q_{X|\tilde{\mathbf{u}}_i}$ using the parameters $\lambda_{X|\tilde{\mathbf{u}}_i}$ and $\mu_{X|\tilde{\mathbf{u}}_i}$. The probability of X transitioning, and not transitioning, to a 1 state by time t_2 is then given by the following two equations.

$$\phi_{\tilde{\mathbf{u}}_i}^{0\to 1}(t) = 1 - \exp(-\lambda_{X|\tilde{\mathbf{u}}_i} t)$$
$$\phi_{\tilde{\mathbf{u}}_i}^{0}(t) = \exp(-\lambda_{X|\tilde{\mathbf{u}}_i} t)$$

Using the available $n+1$ CIMs, we now compute the CIM for any arbitrary assignment of parents $\mathbf{u}$. We calculate the CIMs of X given state instantiations where multiple parents are in state 1 as follows.

$$\begin{aligned}\phi_{\mathbf{u}}^{0}(t) &= \prod_{u_i \in \tilde{\mathbf{u}}^+} \phi_{\tilde{\mathbf{u}}_i}^{0}(t) \\ &= \prod_{u_i \in \tilde{\mathbf{u}}^+} \exp(-\lambda_{X|\tilde{\mathbf{u}}_i} t)\end{aligned}$$

By taking 1 minus this probability, we are left with the probability of transitioning to a state of 1 by time t_2 is reached given any instantiation of parents $\mathbf{u}$.

$$\begin{aligned}\phi_{\mathbf{u}}^{0\to 1}(t) &= 1 - \phi_{\mathbf{u}}^{0}(t) \\ &= 1 - \prod_{u_i \in \tilde{\mathbf{u}}^+} \phi_{\tilde{\mathbf{u}}_i}^{0}(t) \\ &= 1 - \prod_{u_i \in \tilde{\mathbf{u}}^+} \exp(-\lambda_{X|\tilde{\mathbf{u}}_i} t)\end{aligned}$$

This probability allows us to calculate the intensities for the matrix $Q_{X|\mathbf{u}}$, where $\mathbf{u}$ is some instantiation of the parent set $\mathbf{U}$ and more that one variable has a value of 1. Recalling that $\phi_{\mathbf{u}}^{0\to 1}(t) = 1 - \exp(-\lambda_{X|\mathbf{u}} t)$, we can solve for $\lambda_{X|\mathbf{u}}$ and use it to populate the corresponding CIM.

$$\begin{aligned}1 - \exp(-\lambda_{\{X|\mathbf{u}\}} t) &= \phi_{\mathbf{u}}^{0\to 1}(t) \\ 1 - \exp(-\lambda_{\{X|\mathbf{u}\}} t) &= 1 - \prod_{u_i \in \tilde{\mathbf{u}}^+} \exp(-\lambda_{\{X|\tilde{\mathbf{u}}_i\}} t) \\ \exp(-\lambda_{\{X|\mathbf{u}\}} t) &= \prod_{u_i \in \tilde{\mathbf{u}}^+} \exp(-\lambda_{\{X|\tilde{\mathbf{u}}_i\}} t) \\ -\lambda_{\{X|\mathbf{u}\}} t &= \sum_{u_i \in \tilde{\mathbf{u}}^+} -\lambda_{\{X|\tilde{\mathbf{u}}_i\}} t \\ \lambda_{\{X|\mathbf{u}\}} t &= t \sum_{u_i \in \tilde{\mathbf{u}}^+} \lambda_{\{X|\tilde{\mathbf{u}}_i\}}\end{aligned}$$

Note that the t on the left side of the equation is equivalent to the t on the right, indicating that the equality holds for any and all times t in the future. Given this, we can treat t as a constant, resulting in the following.

$$\lambda_{\{X|\mathbf{u}\}} = \sum_{u_i \in \tilde{\mathbf{u}}^+} \lambda_{\{X|\tilde{\mathbf{u}}_i\}} \quad (1)$$

We see that the parameter $\lambda_{\{X|\mathbf{u}\}}$ can be calculated simply by summing the individual $\lambda_{\{X|\tilde{\mathbf{u}}_i\}}$ terms for each parent u_i in $\mathbf{U}$ that is in state 1. The same process can be used to derive the parameter $\mu_{\{X|\mathbf{u}\}}$ for any CIM.

$$\mu_{\{X|\mathbf{u}\}} = \sum_{u_i \in \tilde{\mathbf{u}}^+} \mu_{\{X|\tilde{\mathbf{u}}_i\}} \quad (2)$$

Approximate Inference with Noisy-OR Nodes

Importance sampling can be used to achieve inference over a CTBN that contains Noisy-OR nodes. This works by sampling trajectories from a proposal distribution P' that is guaranteed to conform to evidence. The difference between the proposal distribution P' and the true distribution P is accounted for by weighting each sample by its likelihood.

Trajectories σ can be broken into segments $\sigma[i]$ that are partitioned based on when variables transition between states. The likelihood of a trajectory is decomposable by time and is, therefore, calculated by multiplying the likelihood contributions for each segment of the trajectory. The likelihood for a trajectory is as follows:

$$L'(\sigma) = \prod_{u} \prod_{x} (q_{x|\mathbf{u}}^{M[x|\mathbf{u}]} \exp(-q_{x|\mathbf{u}} T[x|\mathbf{u}]) \prod_{x' \neq x} \theta_{xx'|\mathbf{u}}^{M[x,x'|\mathbf{u}]}$$

where $T[X|\mathbf{u}]$ is the amount of time spent in state x, $M[x,x'|\mathbf{u}]$ is the number of transitions from x to x', $M[x|\mathbf{u}]$ is the total number of transitions from x to any other state. The CIM parameters q and θ are associated with T and M respectively. These sufficient statistics are easily obtained from the trajectory segment. $T[X|\mathbf{u}]$ is simply the duration of the segment, and $M[x|\mathbf{u}]$ is guaranteed to be zero since there are no transitions in the segment by construction. Given this, the likelihood can be rewritten specifically for a trajectory statement.

$$\begin{aligned}\tilde{L}'(\sigma) &= \prod_{\mathbf{u}} \prod_{x} (q_{x|\mathbf{u}}^{0} \exp(-q_{x|\mathbf{u}}(t_e - t)) \prod_{x' \neq x} \theta_{xx'|\mathbf{u}}^{0} \\ &= \prod_{\mathbf{u}} \prod_{x} (\exp(-q_{x|\mathbf{u}}(t_e - t))\end{aligned}$$

Introducing Noisy-OR nodes to the network requires a slight adjustment to the sampling process, as there is no longer any guarantee that the rate parameters necessary for the algorithm will be precomputed. The rates are necessary when drawing from the exponential or truncated exponential distributions, as well as when computing the likelihoods for the weights. We extend importance sampling by using Algorithm 1 to obtain rates whenever they are required.

In the event the requested rate parameter comes from an instantiation of the parents such that no more than one parent is in state 1 at a time, then the rate parameter is retrieved

Algorithm 1 Retrieve Intensity

1: **procedure** GET-RATE($x, \mathbf{u}$)
2: $\quad q_{x|\mathbf{u}} \leftarrow 0$
3: $\quad n \leftarrow \sum_{u_i \in \mathbf{u}} u_i$ ▷ Count parents in state 1
4: $\quad$ **if** $n \leq 1$ **then**
5: $\quad\quad q_{x|\mathbf{u}} \leftarrow Q_{X|\mathbf{u}}[x, x]$
6: $\quad$ **else**
7: $\quad\quad$ **for** $u_i \in \mathbf{u}$ **do**
8: $\quad\quad\quad$ **if** $u_i = 1$ **then**
9: $\quad\quad\quad\quad q_{x|\mathbf{u}} \leftarrow q_{x|\mathbf{u}} + Q_{X|u_i}[x, x]$
10: $\quad$ **return** $q_{x|\mathbf{u}}$ ▷ The intensity

from the corresponding intensity matrix. If the rate is due to multiple parent nodes being in state 1, then no such intensity matrix exists and the intensity must be computed on the fly. The generated rate parameter is calculated by summing the corresponding rates from each CIM where the parent was in state 1. This is shown in the algorithm on line 9, which implicitly makes use of Equations 1 and 2. Note that there is a potential increase in the computational complexity that is linear in the number of parents when retrieving rates from CIMs. Fortunately, the model can be represented with fewer CIMs, which may reduce the need to retrieve information from disk and save on expensive I/O operations.

Noisy-OR Expectation Comparison

In this section, we discuss experiments designed to validate and explore the behavior of the proposed Noisy-OR parameterization. We compare the query results of a network parameterized using Noisy-OR with the expected probabilities to show equivalence. This experiment is designed to further justify the formulation of Noisy-OR in CTBNs by supporting the theoretical results already provided.

We constructed a network with three parents A, B and C, and a single child node X as shown in Figure 2. Each of the parents is parameterized with an initial distribution of $(0.5, 0.5)$ and a single CIM containing a rate of 1.0 for transitioning to either state 0 or 1. The child node X is parameterized with an initial distribution of $(1.0, 0.0)$, and uses the Noisy-OR model. This means that only four CIMs are specified for X rather than the full eight. The rate of transitioning to state 1 given that no parents are in state 1 is 0.0. When only A is in state 1, the rate is 0.75, only B is 0.85, and only C is 0.8. For each CIM, the rate of transitioning from state 1 to state 0 is 0.0.

In the event that multiple parents enter state 1, the intensities are not specified and will need to be computed. The intensities are computed by summing the corresponding intensities from the CIMs that *are* specified. The expectation is that the resulting CIMs will cause a transition after some time t in the future with a probability equal to the product of the probabilities that would occur according to the specified CIMs. To verify that this behavior is achieved, we compare the probabilities obtained by querying the network with the expected probability computed via multiplication.

Evidence was applied for the four cases where multiple parents were in state 1 over the interval of time $t = [0.0, 2.0)$. The probability of X being in state 1 was queried at 100 evenly spaced timesteps over this time interval. The expected probability was computed as the product of the probabilities for the cases where individual parents are in state 1. The mean differences for all timesteps are shown in Table 2. A paired equivalence test with a 0.005 region of similarity and a significance level of 0.05 was used to compare the queried result to the expected probability. For all cases, it was found that the probability obtained by querying the Noisy-OR parameterized CTBN was equivalent to the corresponding expected probability.

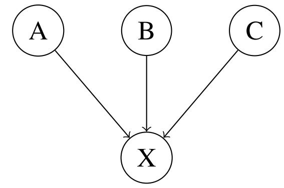

Figure 2: Network structure used in the expectation comparison experiment. X uses a Noisy-OR parameterization.

	Queried	Expected
$P(X\|AB)$	0.6876482	0.6865322
$P(X\|BC)$	0.6956636	0.6963848
$P(X\|AC)$	0.7031165	0.7033511
$P(X\|ABC)$	0.7896124	0.7885470

Table 2: The Queried column shows the probability of X being in state 1 given the four combinations of A, B, and C being in state 1 as returned by inference, which requires CIMs generated on the fly. This can be compared to the Expected column, which shows the exact value computed manually.

Applications

The Noisy-OR model is useful for any scenario where the parents of a variable have disjunctive interactions, for example, when the model can be interpreted as causal. In these networks, the state of a parent is considered to be an event that causes the state of a child node, which is viewed as an effect. Disjunctive interaction occurs when any parent on its own is sufficient to explain the behavior of the child.

Due to its compatibility with causal networks, Noisy-OR has been used in BNs for the purpose of diagnostics. These diagnostic models consist of a set of faults and a set of tests. Using a standard BN parameterization, the CPT for a test node is exponential in the number of faults that it monitors. Fortunately it is often the case that the occurrence of any fault is sufficient to cause a test to fail. This disjunctive interaction between the faults allows for a Noisy-OR parameterization of the tests, reducing the size of the CPT to be linear in the number of monitored faults.

These diagnostic models have proved useful in the medical domain, where faults in the network are considered to be a disease, and tests are referred to as findings. One such diagnostic model that has been studied in the literature is the

Quick Medical Reference, Decision- Theoretic (QMR-DT) network (Shwe et al. 1991). Heckerman demonstrated the effectiveness of the Noisy-OR parameterization in BNs by applying it to the QMR-DT model (Heckerman 1990).

CTBNs allow us to look beyond diagnostic models, and instead focus on prognostic models that are capable of predicting faults that may occur at some time in the future. As an example, the QMR-DT model could be extended to perform prognostics using a CTBN with the same network structure. By adapting the Noisy-OR model for CTBNs, it is now possible to model real-world temporal problems that would have otherwise been intractable.

Conclusion

We have described how the Noisy-OR model works in the context of CTBNs. This parameterization reduces the number of specified CIMs to be linear in the number of parents rather than exponential. It was shown how to calculate the unspecified rates for CIMs where multiple parents are in state 1 by summing the rates from specified CIMs. Finally, inference involving Noisy-OR nodes was demonstrated by using a modified version of importance sampling. These results were then validated experimentally.

Note that while the Noisy-OR model reduces the number of CIMs that must be specified, the complexity of inference remains the same. This is evident by the `Get-Rate` procedure, which allows importance sampling to operate as if the full set of CIMs were available. Future work will focus on exploiting the Noisy-OR structure to reduce inference complexity. Another area of interest is the automatic identification of potential Noisy-OR nodes. Existing work has been done on sensitivity analysis for CTBNs to identify how changes to parameterization affect the resulting model behavior (Sturlaugson and Sheppard 2015). Sensitivity analysis could be used to compare existing nodes in a CTBN to the corresponding Noisy-OR formulation to identify potential candidates for conversion to the Noisy-OR parameterization. Finally we would like to generalize this Noisy-OR model to permit non-binary states and additional functions aside from the OR gate.

References

Bobbio, A.; Portinale, L.; Minichino, M.; and Ciancamerla, E. 2001. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. *Reliability Engineering & System Safety* 71(3):249–260.

Cao, D. 2011. *Novel models and algorithms for systems reliability modeling and optimization*. Ph.D. Dissertation, Wayne State University.

Diez, F. J. 1993. Parameter adjustment in Bayes networks. the generalized noisy or-gate. In *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence*, 99–105. Morgan Kaufmann Publishers Inc.

Heckerman, D., and Breese, J. S. 1994. A new look at causal independence. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, 286–292. Morgan Kaufmann Publishers Inc.

Heckerman, D., and Breese, J. 1996. Causal independence for probability assessment and inference using Bayesian networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 26(6):826–831.

Heckerman, D. 1990. A tractable inference algorithm for diagnosing multiple diseases. *Uncertainty in Artificial Intelligence* 5(1):163–171.

Heckerman, D. 1993. Causal independence for knowledge acquisition and inference. In *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence*, 122–127. Morgan Kaufmann Publishers Inc.

Henrion, M. 1987. Practical issues in constructing a Bayes' belief network. *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence* 132–139.

Johnson, N. L. 1999. *Survival models and data analysis*, volume 74. John Wiley & Sons.

Kleinbaum, D. G., and Klein, M. 2012. Parametric survival models. In *Survival analysis*. Springer. 289–361.

Murphy, K., and Mian, S. 1999. Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA.

Nodelman, U.; Shelton, C. R.; and Koller, D. 2002. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 378–387. Morgan Kaufmann Publishers Inc.

Oniśko, A.; Druzdzel, M. J.; and Wasyluk, H. 2001. Learning Bayesian network parameters from small data sets: Application of noisy-or gates. *International Journal of Approximate Reasoning* 27(2):165–182.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

Perreault, L.; Thornton, M.; Goodman, R.; and Sheppard, J. 2015. Extending continuous time Bayesian networks for parametric distributions. In *IEEE Swarm Intelligence Symposium*.

Shwe, M. A.; Middleton, B.; Heckerman, D.; Henrion, M.; Horvitz, E.; Lehmann, H.; and Cooper, G. 1991. Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base. *Methods of information in Medicine* 30(4):241–255.

Srinivas, S. 1993. A generalization of the noisy-or model. In *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence*, 208–215. Morgan Kaufmann Publishers Inc.

Strasser, S., and Sheppard, J. 2013. An empirical evaluation of Bayesian networks derived from fault trees. In *IEEE Aerospace Conference, 2013*, 1–13.

Sturlaugson, L., and Sheppard, J. 2015. Sensitivity analysis of continuous time Bayesian network reliability models. *SIAM/ASA Journal on Uncertainty Quantification* 3(1):346–369.