Extending Continuous Time Bayesian Networks

Karthik Gopalratnam and **Henry Kautz** and **Daniel S. Weld**
Department of Computer Science
University of Washington
Seattle, WA-98195-2350
{karthikg,kautz,weld}@cs.washington.edu

Abstract

Continuous-time Bayesian networks (CTBNs) (Nodelman, Shelton, & Koller 2002; 2003), are an elegant modeling language for structured stochastic processes that evolve over continuous time. The CTBN framework is based on homogeneous Markov processes, and defines two distributions with respect to each local variable in the system, given its parents: an exponential distribution over *when* the variable transitions, and a multinomial over *what* is the next value. In this paper, we present two extensions to the framework that make it more useful in modeling practical applications. The first extension models arbitrary transition time distributions using Erlang-Coxian approximations, while maintaining tractable learning. We show how the censored data problem arises in learning the distribution, and present a solution based on expectation-maximization initialized by the Kaplan-Meier estimate. The second extension is a general method for reasoning about negative evidence, by introducing updates that assert no observable events occur over an interval of time. Such updates were not defined in the original CTBN framework, and we show show that their inclusion can significantly improve the accuracy of filtering and prediction. We illustrate and evaluate these extensions in two real-world domains, email use and GPS traces of a person traveling about a city.

Introduction

Many problems in artificial intelligence involve reasoning about complex stochastic systems that evolve over time. Dynamic Bayesian networks (DBNs) (Dean & Kanazawa 1989) are a popular approach, which provided a factored representation of discrete-time processes. Unfortunately, DBNs use a discrete temporal representation, which is awkward and computationally expensive when the absolute time of events is important and observations occur irregularly. In such a case the granularity of each DBN time slice must correspond to the smallest possible interval between observations. For example, if observations are usually separated by hours but occasionally by a second, then a DBN update must be performed every second, unless the absolute time of the observations is to be ignored.

Continous-time Markov chains provide a way of describing discrete-state systems that evolve according to exponential time distributions, and are widely used in operations research, astronomy, and biology. The most common form, called a homogenous Markov process, is represented by a matrix that is quadratic in the size of the state space. A homogenous Markov process can be projected to any real-valued future time point with a single matrix exponential operation, so updates by projection and conditioning on observations are similarly efficient.

Nodelman *et al.* [2002] introduced *continuous time Bayesian networks* (CTBNs), which are a *factored* representation of continuous-time Markov chains. As with other graphical representations, CTBN can provide a compact representation of a domain by making explicit many of the conditional independence relationships between variables. Also like other factored representations such as Bayesian networks or DBN's, the small size of the representation is no guarantee that exact inference is tractable. Indeed, the only known exact inference method for CTBNs is compilation to an explicit homogenous Markov process. Nodelman *et al.* present an approximate inference method based on the clique tree inference algorithm. Space precludes description of our own particle filtering approximation algorithm.

Beyond issues of worst-case complexity (which affect all probabilistic representations), CTBN's have two significant limitations: First, CTBNs are limited to modeling processes with exponential time distributions, but other distributions frequently occur in practice. Second, the only update method in the original formulation of CTBNs is based on the value of a variable at a single point in time. However, in many systems sensors operate continuously, reporting only when a change occurs. For example, a motion detector is silent until it senses activity and a phone rings only when an incoming call is present. When modeling these systems it is essential to reason about evidence of the form "variable X stayed at value x for the entire interval $[t_1, t_2]$." Because of the continuous nature of time, such interval updates are *not* equivalent to a finite series of point-based updates. This second limitation is an example of the general negative evidence problem that arises in modeling any dynamic system: that is, how to efficiently incorporate observations of the myriad possible events that did *not* occur.

In this paper, we resolve these limitations, presenting two extensions to the CTBN framework.

- We show how to model arbitrary transition time distributions using Erlang-Coxian approximations, while maintaining tractable learning. We show how the censored data problem arises in learning time distributions, and present

a solution based on expectation-maximization initialized by the Kaplan-Meier estimate.

- We present a general method for reasoning about negative evidence, by introducing updates that assert no observable events occur over an interval of time. We demonstrate that interval evidence updates can significantly improve the accuracy of filtering and prediction.

We illustrate and evaluate these extensions in two real-world domains. The first is a model of person's use of email, which can be used to predict when the person will reply to an incoming message. The second models the pattern of person's movements through a city using various modes of transportation. The model is trained on a log of GPS (global positioning system) data, and can be used to predict the time it takes for a user to reach locations of interest, etc.

Continuous-Time Bayesian Nets (CTBNs)

In this section, we summarize certain key notions about continuous time Bayesian networks (CTBNs) presented in (Nodelman, Shelton, & Koller 2002). A CTBN represents a stochastic process over a structured state space consisting of assignments to a set of local variables. CTBNs describe the dynamics of the temporal evolution of this structured state space in terms of the dependencies among the evolution of the local variables as follows.

Homogeneous Markov Process

Let X be a state variable with finite domain $Val(X) = \{x_1, \ldots, x_n\}$ whose value changes continuously over time. We may define the dynamics of this system in terms of a *homogeneous Markov process*, $X(t)$, by defining its *intensity matrix* (IM) as follows:

$$\vec{\boldsymbol{Q}}_{\boldsymbol{x}} = \begin{bmatrix} -q_1^x & q_{12}^x & \cdots & q_{1n}^x \\ q_{21}^x & -q_2^x & \cdots & q_{2n}^x \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1}^x & q_{n2}^x & \cdots & -q_n^x \end{bmatrix}$$

where $q_i^x = -\sum_{j \neq i} q_{ij}^x$. Intuitively, the diagonial elements specify how long the system will remain in a state, and the other elements specify the probability of transitioning between states.

Specifically, the probability density function for the time spent in a state x_i is $f(t) = q_i^x e^{-q_i^x t}$, with corresponding distribution function $F(t) = 1 - e^{-q_i^x t}$. When the X leaves state x_i, it will transition to x_j with probability q_{ij}^x / q_i^x. We can project a initial probability distribution described by a vector P_X^0 to a future point t by computing $P_X(t) = P_X^0 \text{expm}(\vec{\boldsymbol{Q}}_{\boldsymbol{x}} t)$. The function $expm(.)$ refers to matrix exponentiation, and is the mathematical operation that considers all possible *trajectories* of X through its state space up to time t.

Subsystems of Markov Processes A *subsystem* S describes the behavior of the process over a subset of the full space, *i.e.* $Val(S) \subset Val(X)$. The intensity matrix of S, $\boldsymbol{Q}_S$, can be formed by considering only those entries from $\boldsymbol{Q}_X$ that correspond to states in S. Since in general a subsystem is not closed, we can form queries about when the system the will enter or exit the subsystem, and the amount of time spent there. This last quantity, called the *holding time*, has the distribution $F(t) = 1 - P_S^0 expm(\boldsymbol{Q}_S t)\vec{\boldsymbol{e}}$, where P_S^0 is the entrance distribution and $\vec{\boldsymbol{e}}$ is a column vector of ones.

Continuous Time Bayesian Networks

The joint dynamics of several local variables is captured in the notion of the *conditional Markov process*, which forms the basic building block of the CTBN framework. A conditional Markov process Y is an *inhomogeneous* Markov process whose IM varies as a function of the values of a set of conditioning variables $\vec{\boldsymbol{V}}$, referred to as the parents of Y. For each instantitation of values $\vec{\boldsymbol{v}}$ to $\vec{\boldsymbol{V}}$ the variable Y is governed by an an intensity matrix $\boldsymbol{Q}_{Y|\vec{\boldsymbol{v}}}$. The full set of matrices for Y is called its *conditional intensity matrix* (CIM) $\boldsymbol{Q}_{Y|\vec{\boldsymbol{V}}}$. A CTBN is then formed by composing together these local conditional processes. A graphical structure encoding the dependencies between the variables in the system, together with an initial distribution over this state space, and the local CIMs for each variable completely specify the CTBN.

The semantics of CTBNs can be understood in two terms. The first is based on viewing the entire system as a composite IM describing a homogeneous Markov process over the joint entire state space via an *amalgamation* operation which combines all the local CIM to produce the Joint Intensity Matrix. The evolution of the CTBN is then completely specified by this Joint Intensity Matrix. The other is based on a *generative* perspective where the CTBN is viewed as defining a generative model over a set of events which correspond to variables in the system taking on certain values at specific times.

Erlang-Coxian CTBNs (EC-CTBNs)

Because CTBNs rely on well-behaved exponential distributions for modeling temporal distributions, they submit to easy analytic treatment. Unfortunately, many real-world distributions can not be modeled accurately with exponentials, thus we seek a representation which combines the benefits of CTBNs with greater expressive power.

Our solution is to use *phase-type* (PH) distributions (Neuts 1981), which also obey the Markov property and hence maintain analytical tractability. Phase-type distributions have been widely used to closely approximate general distributions and hence provide great expressiveness. Specifically, we employ the *Erlang-Coxian* (EC) distribution, which has the attractive property that it is described by a small number of free parameters which are easily computed in closed form.

Phase-Type Distributions

An n^{th}-order PH distribution is specified in terms of the absorption time of a corresponding Markov chain consisting of n states (*phases*), where the i-th phase has an exponentially distributed holding time with rate λ_i. To fully specify this process one must dictate the *entrance distribution*, $\vec{\boldsymbol{\tau}} = [p_{01} \ldots p_{0n}]$, where p_{0i} denotes the probability that the chain starts in phase i, and an $n \times n$ *infinitesimal generator matrix*, $\vec{\boldsymbol{T}}$, whose entries, p_{ij}, encode the probability of transitioning from phase i to phase j. The cumulative distribution function is $F(t) = 1 - \vec{\boldsymbol{\tau}} \text{expm}(\boldsymbol{T}t)\vec{\boldsymbol{e}}$. The i-th

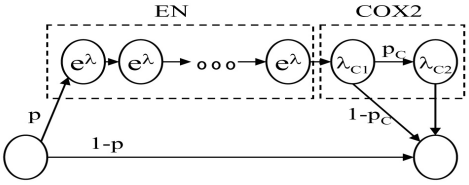

Figure 1: Markov chain for an Erlang-Coxian distribution

component of the chain's *exit distribution*, $\vec{\boldsymbol{u}} = \vec{\boldsymbol{T}} \cdot \vec{\boldsymbol{e}}$, is the probability that the final absorbing state is reached by transitioning from the chain's i-th phase.

For many years, researchers have devised methods for approximating an arbitrary distribution, G, with a phase-type distribution PH. We make the common assumption that PH is a good match to G when its first k moments equal those of G. Thus we wish to find values for $\vec{\boldsymbol{\tau}}$ and $\vec{\boldsymbol{T}}$ such that the first k moments match. Note that this search space is very large, since we must find values for $\Theta(n^2)$ parameters, when considering PH-distributions of order n.

Recently, (Osogami & Harchol-Balter 2003b; 2003a) published an efficient and elegant approximation method. Their approach is efficient because it restricts the search to the space of *Erlang-Coxian* (EC) distributions — distributions whose Markov chain consists of $n-2$ phase Erlang distribution followed by a two-phase Coxian$^+$ distribution as specified in Figure 1. Note that the rates λ are the same for all Erlang phases, and that the only way to exit the chain is via the first phase of the Erlang distribution or one of the two Coxian phases. The EC distribution is therefore completely described by just six free parameters, $(n, p, \lambda, \lambda_{C1}, \lambda_{C2}, p_C)$, which can be calculated in closed form given the first three moments of any distribution G.

Generalized Markov Processes

Let X be a random variable governed by a Markov process whose transition intensities may vary over time (*i.e.*, nonhomogenous). We can model $X(t)$ with a *generalized Markov process*, $\hat{X}$, which is a homogeneous Markov process over an extended state space. We construct the generalized process by approximating each holding-time distribution of the original process with an EC-distribution. Each atomic transition of the original process $X(t)$ is then represented by a *subsystem* of $\hat{X}$.

As a first step, let G_{x_i} denote the distribution over holding times for each value $x_i \in \mathit{Val}(X)$ in the original process. We now approximate each G_{x_i} with an EC distribution represented by a chain with phases $S_{x_i} = \{x_1^i, \cdots, x_{N_i}^i\}$, generator matrix $\vec{\boldsymbol{T}}_{x_i}$ and exit distribution $\vec{\boldsymbol{u}}_{x_i}$. The (homogeneous) generalized Markov process for X is defined as a Markov process over $\hat{X}$, where $\mathit{Val}(\hat{X}) = \bigcup_{i=1}^{n} S_{x_i}$. The number of states in this new process is $N_{\hat{X}} = |Val(\hat{X})| = \sum_i^n N_i$. The dynamics of the generalized process are governed by $\vec{\boldsymbol{Q}}_{\hat{\boldsymbol{X}}}$, an $N_{\hat{X}} \times N_{\hat{X}}$ matrix, called the *generalized intensity matrix* (GIM), which is computed by amalgamating the Markov chains $\vec{\boldsymbol{T}}_{x_i}$ and weighting the transitions between these subsystems by the exit distributions from the respective chains.

In other words, $\vec{\boldsymbol{Q}}_{\hat{\boldsymbol{X}}}$ contains n subsystems — one for each possible value $x_i \in \mathit{Val}(X)$. We call these original values of X, the *base values* of the generalized Markov process $\hat{X}$. By construction, the GIM has semantics such that $(\hat{X} = x_k^i) \implies (X = x_i)$, where x_k^i is the k^{th} state in the underlying Markov chain for $X = x_i$.

The GIM can now be queried in the same way that a simple intensity matrix can, *e.g.*, in order to determine the distribution over the states in time. Given an initial distribution $\hat{P}_0$, the new distribution over the states of $\hat{X}$ is given by $\hat{P}_t = \hat{P}_0 \text{expm}(\vec{\boldsymbol{Q}}_{\hat{\boldsymbol{X}}} t)$. Note that this posterior distribution is over all the states of the extended process. The probability of a particular base value $P_t(X = x_i) = \sum_{j=1}^{N_i} \hat{P}_t[x_j^i]$.

The EC-CTBN Model

We can compose generalized Markov processes in the same manner as we did for conditional Markov processes above. Let $Y(t)$ be a generalized Markov process whose dynamics are conditioned on a set of other variables $\mathcal{V}$ that themselves evolve as generalized Markov processes. Then a *generalized conditional intensity matrix* (GCIM) for Y is a set of generalized intensity matrices, one for each instantiation of *base values* to the conditioning variables $V_i \in \mathcal{V}$. GCIMs enable us to model the local dependence of one variable on a set of other variables, in the same way that CIMs did for CTBNs.

We now define EC-CTBNs. Let $\mathcal{X} = \{X_1, \ldots, X_k\}$ be a set of discrete random variables. An *Erlang-Coxian continuous-time Bayesian network* (EC-CTBN) is a triple $\langle P^0_{\mathcal{X}}, \mathcal{G}, \mathcal{Q} \rangle$. $P^0_{\mathcal{X}}$ is a factored probability distribution over the initial values of the X_i. $\mathcal{G}$ is a directed graph with whose nodes are $\mathcal{X}$, and $\mathit{Par}(X_i)$ denotes the parents of X_i in $\mathcal{G}$. $\mathcal{Q}$ is a set of generalized conditional intensity matrices containing $\vec{\boldsymbol{Q}}_{\boldsymbol{X_i}|\mathit{Par}(\boldsymbol{X_i})}$ for each $X_i \in \mathcal{X}$.

The global semantics of the EC-CTBN are understood in terms of the amalgamation operation over the GCIMs, which define the EC-CTBN. The Joint Intensity Matrix defined by this operation represents a *Homogeneous* Markov Process over the extended state space that includes the states corresponding to the phases of the various EC-chains. However, the evolution of the probability distributions over the extended state space, when projected onto the corresponding base values, reflect the different distributions over the holding times of the variables in their base states.

Learning EC-CTBNs

Maximum likelihood estimation (MLE) of CTBN parameters is efficient due to the special nature of the exponential distribution. As described in (Nodelman, Shelton, & Koller 2003), the parameters for a variable X can be calculated from two sets of simple statistics: $T[x|\vec{\boldsymbol{v}}]$, the total amount of time X spent in state x, conditioned on its parent variables taking on the value $\vec{\boldsymbol{v}}$; and $M[x, x'|\vec{\boldsymbol{v}}]$, the total number of times X transitioned from x to x' for each pair of states. Then the MLE parameters of the model are given by $\hat{q}_{x|\vec{\boldsymbol{v}}} = M[x|\vec{\boldsymbol{v}}]/T[x|\vec{\boldsymbol{v}}]$ and $\hat{q}_{xx'|\vec{\boldsymbol{v}}} = M[x, x'|\vec{\boldsymbol{v}}]/T[x|\vec{\boldsymbol{v}}]$, where $M[x|\vec{\boldsymbol{v}}] = \sum_{x'} M[x, x'|\vec{\boldsymbol{v}}]$.

Extending the expressive power of CTBNs to handle general non-exponential time distributions using hidden state is

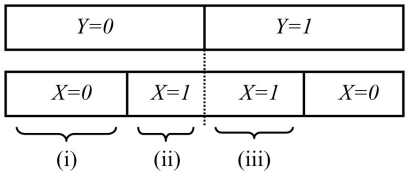

Figure 2: Interval (i) counts toward $T[X=0 \mid Y=0]$; (ii) is a censored interval associated with $T[X=1 \mid Y=0]$; (iii) counts toward $T[X=1 \mid Y=1]$.

challenging due to the infinitely-large space of possible trajectories of the hidden variable. In the special case of EC distributions, these trajectories are captured by the Markov chain described in closed form by the six parameters described above. Thus, at first glance, the problem of learning EC-CTBNs might appear to be solved, because the six parameters can be estimated from the first three moments of the data.

However, a problem arises with this simple approach, because the time that a variable spends in a state may be governed by more than one instantiation of its parents. Consider the example of two Boolean variables X and Y, where $Y = \mathit{Par}(X)$. Suppose X switches to the value 1 while Y is 0. After a while Y switches to 1 and X remains unchanged, as shown in Figure 2. The problem is how to account for the measurement of the interval (ii). It cannot be directly counted in $T[X = 1 \mid Y = 0]$, because it ends when Y switched, not because X switched. Nor can the union of intervals (ii) and (iii) be directly counted toward $T[X=1 \mid Y=0]$. Furthermore, discarding measurements of this form could lead to an arbitrarily-poor learned model, since the observation *does* tell us that our estimate of $\hat{q}_{X=1|Y=0}$ should be *constrained* to account for the fact that in this case X did *not* switch before Y. Intuitively, we should adjust the length of (iii) to be what it would of have been if Y had not switched first.[1]

The problem of handling such truncated, or *censored*, data is studied in the field of failure analysis (Crowder *et al.* 1994). While classic failure analysis has been concerned with fitting censored data to simple distributions (such as the normal, exponential, Weibull, *etc.*), there has been some work on fitting phase-type distributions of fixed order to censored data using expectation-maximization (EM) (Asmussen, Nerman, & Olsson 1996). We generalize this approach to fit EC distributions of arbitrary order as follows.

We record the training data D_X for each variable X as a set of tuples $\langle x, \vec{\boldsymbol{v}}, t, c\rangle$, where x is a value of X, $\vec{\boldsymbol{v}}$ are instantiations of the parents of X, t is the length of the interval, and $c = 0$ if the interval ends with a transition of X, or $c = 1$ (censored) if it ends with a transition of a parent of X. We then assign the weight of each censored element in D_H evenly among all the longer non-censored elements (a technique known as the Kaplan-Meier estimate), and calculate an initial estimate of the parameters of the best EC approximation of the weighted, non-censored data using Osogami & Harchol-Balter's algorithm. Then, in standard EM fashion, the current EC approximation is used to re-estimate the values of censored data points, and the process repeats until convergence. The final result is used to define the generalized intensity matrix $\vec{\boldsymbol{Q}}_{\hat{\boldsymbol{X}}}$.

In order to avoid having to re-design the Osogami & Harchol-Balter procedure to accept as input probability distributions over possible values for the censored data, one can employ a Monte-Carlo version of EM, where the expectation step draws a number of samples for each censored point. In fact, in the experiments described below we found that EM converged to same values even if each censored point was simply replaced by its analytically derived expected value, which led to a very efficient implementation.

Handling Evidence over Time Intervals

The second major drawback of CTBNs is their inability to model *interval evidence*, *i.e.* evidence of the form "Variable X stayed at value x_i during time interval $[t_1, t_2]$." Interval evidence arises frequently in a variety of practical situations. For example, a motion detector might report that no one entered a room between 8:30 and 10:00AM. CTBN's inability stems from the semantics of homogeneous Markov process evolution, which is described in terms of the exploration of the *trajectories* of variables during a given interval. The posterior distribution $P_t = P_0 \text{expm}(\vec{\boldsymbol{Q}}t)$ for an intensity matrix $\vec{\boldsymbol{Q}}$ includes the effects of an arbitrary number of transitions at an arbitrary number of time points in the interval $[0, t]$. Since this quantification over all trajectories (including those where X temporarily shifted away from x_i) is implicit in use of matrix exponentiation for projection, there appears to be no way to benefit from such evidence.

However, transforming the state space once again solves the problem. Indeed, our solution works on both CTBN and EC-CTBN models. The crux of the issue is the need to measure the probability mass of the subset of trajectories consistent with the evidence — or equivalently, preventing probability mass from accumulating in trajectories which violate the interval evidence. The key insight is that we can enforce the constraint $X = x_i, \forall t \in [t_1, t_2]$ by creating and conditioning on a new variable which records whether the observed variable, *e.g.*, X, changed state during the interval.

Specifically, let $\mathcal{X}$ be a set of discrete random variables containing X. Let ζ be a new Boolean variable; intuitively, $\zeta = 0$ will signify that X did not change value anywhere in a prediction interval, while $\zeta = 1$ indicates that there was at least one transition. In other words, ζ *partitions* the space of all trajectories over $\mathcal{X}$, based on whether or not X transitioned.

We create a new (*augmented*) intensity matrix, $\vec{\boldsymbol{Q}}_{\boldsymbol{\zeta}}$ over $\mathcal{X} \bigcup \{\zeta\}$ as follows. Let $\vec{\boldsymbol{Q}}_{\boldsymbol{\mathcal{X}}}$ be an $n \times n$ intensity matrix for the amalgamated system, and let X be a state variable for which we expect interval evidence. We define $\vec{\boldsymbol{Q}}_{\boldsymbol{\mathcal{X}}}$'s *violation matrix*, $\vec{\boldsymbol{V}}$, to be a $n \times n$ matrix whose entries, v_{ij} are defined as follows. If X has different values in states i and j of the amalgamated Markov process, then v_{ij} equals the ij^{th} element of $\vec{\boldsymbol{Q}}_{\boldsymbol{\mathcal{X}}}$ otherwise $v_{ij} = 0$. Intuitively, $\vec{\boldsymbol{V}}$ records the intensity of all transitions where X might change

[1] This problem doesn't affect CTBNs, because of mathematical properties specific to the exponential distribution. However, this immunity fails when the variables can have arbitrary distributions over holding times. See (Nodelman, Shelton, & Koller 2003).

value, violating the evidence. We now define the $2n \times 2n$ augmented intensity matrix, $\vec{Q}_\zeta$:

$$\vec{Q}_\zeta = \begin{bmatrix} (\vec{Q}_{\boldsymbol{X}} - \vec{V}) & \vec{V} \\ \vec{0} & \vec{Q}_{\boldsymbol{X}} \end{bmatrix}$$

$\vec{Q}_\zeta$'s four quadrants correspond to the transitions between possible values of ζ. In the upper left, ζ remains zero, so transitions which change X are disallowed. The upper right denotes transitions where X changes value for the first time, here ζ also changes from zero to one. The bottom left is all zeros because ζ cannot transition from one to zero. The bottom right is the original intensity matrix since all transitions are allowed once $\zeta = 1$.

For example, consider a simple CTBN with two Boolean nodes, $\mathcal{X} = \{Y, X\}$, where Y is the parent of X, and X is the variable about which we receive interval observations. Let the joint intensity matrix be:

$$\vec{Q}_{XY} = \begin{bmatrix} -3 & 2 & 1 & 0 \\ 2 & -4 & 0 & 2 \\ 1 & 0 & -2 & 1 \\ 0 & 1 & 3 & -4 \end{bmatrix}$$

Then the augmented intensity matrix is

$$\vec{Q}_\zeta = \left[\begin{array}{cccc|cccc} -3 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & -4 & 0 & 2 & 2 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -4 & 0 & 0 & 3 & 0 \\ \hline 0 & 0 & 0 & 0 & -3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & -4 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & -4 \end{array}\right]$$

We can use $\vec{Q}_\zeta$ to propagate an initial distribution P_{t_1} which has $\zeta = 0$. Given the posterior P_{t_2}, we discard any probability mass assigned to $\zeta = 1$ and re-normalize. This maintains the original CTBN semantics, but enforces the fact that X didn't change during the interval.

Note that we need just one ζ variable to handle interval evidence over an arbitrary number of variables;[2] in such a case, the semantics of $\zeta = 0$ is that *none* of the observed variables changed during the interval. Furthermore, note that since our method simply transforms a homogeneous Markov process into one with an additional variable, it may be used with any inference algorithm.

Experiments

Our evaluation addresses two questions. First, does the generality and expressive power of our extension to EC distributions provide benefits on real problems? Second, does our method for handling interval evidence matter, or are the effects of such observations insignificant? To answer these questions, we chose two domains: email response behavior and the day-to-day activity of a human navigating between home, work and other locations as recorded by GPS data.

Our email domain models a email-response sequence with three variables: U denotes user state and has two values *online* and *offline*. Variable, S, conditioned on U, denotes message staleness (intuitively, a measure of how long the message has remained unanswered) and has four values $\{0, 1, 2, \geq 3\}$ which denote the number of times the user has gone offline without replying to the message. The third variable, E, denotes the email-state and has two values *waiting for reply* and *replied*; E is conditioned on both U and S.

[2]Note that while different intensity matrices are needed for different sets of evidence, the $\vec{Q}_\zeta$ can be generated very quickly.

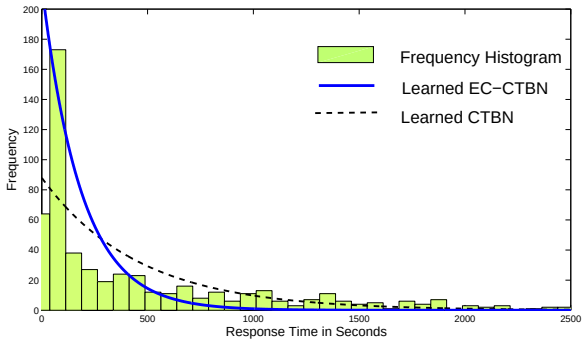

Figure 3: EC-CTBN and CTBN approximation overlayed on email response-time data.

Given three months worth of the author's actual email, we found 533 email-response pairs. We represented each of these examples as a variable-length sequence of events (*e.g.*, message arrival, user goes offline, and user replies). We estimated values for U at the time of these events based on his proximal email activity. Then using 5-fold cross validation, we split the data into training sets of 433 and test sets of 100 message-response episodes.

In order to evaluate EC-CTBN efficacy, we learned a CTBN and a EC-CTBN for the network mentioned above. (The learned EC-CTBN had 288 states compared to the CTBN's 16.) We then queried the two models for the probability density over reply events in the hold-out set. Figure 3 shows the EC-CTBN and (traditional) CTBN density function over these reply-times mapped on top of the actual frequency distribution for a representative hold-out set. Visually, it is apparent that the EC-CTBN does a better job of matching the actual distribution. We confirmed this quantitatively by computing the ratio of the probability densities predicted by the two models for every reply in the hold-out set. Averaging the mean over 5 runs gave a value of 1.38, *i.e.* the EC-CTBN was almost forty percent more likely to predict the correct answer than the ordinary CTBN.

In our second test, we evaluated the effect of our technique for handling negative evidence in the form of interval observations. Here we compared the learned CTBN model to an augmented version (described in the previous Section) of the same CTBN model. We tested on the same hold-out set as in the last experiment, but incorporated evidence of the form "User was online from t_i to t_j." Figure 4 shows that the likelihood of the data under the CTBN augmented with interval evidence is substantially higher than a CTBN without interval evidence. The latter model predicts a longer response time, since it erroneously thinks the user may be offline and hence less responsive. Similar experiments show an augmented EC-CTBN outperforms a standard EC-CTBNs on the same data.

We next consider a different domain: a month's worth of GPS trace data, recording an individual's movement about

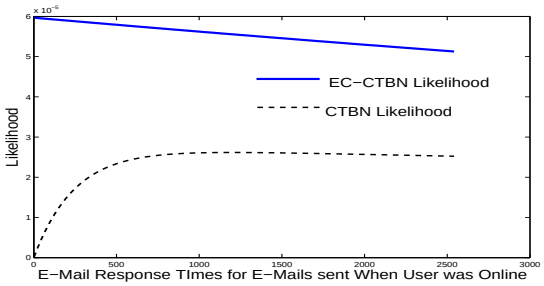

Figure 4: Average likelihood of evidence *vs.* the time to reply to email. (The Y axis measures likelihood times 10^{-5}.) Since it can exploit interval evidence, the augmented CTBN predictions are much more accurate than those of the initial CTBN.

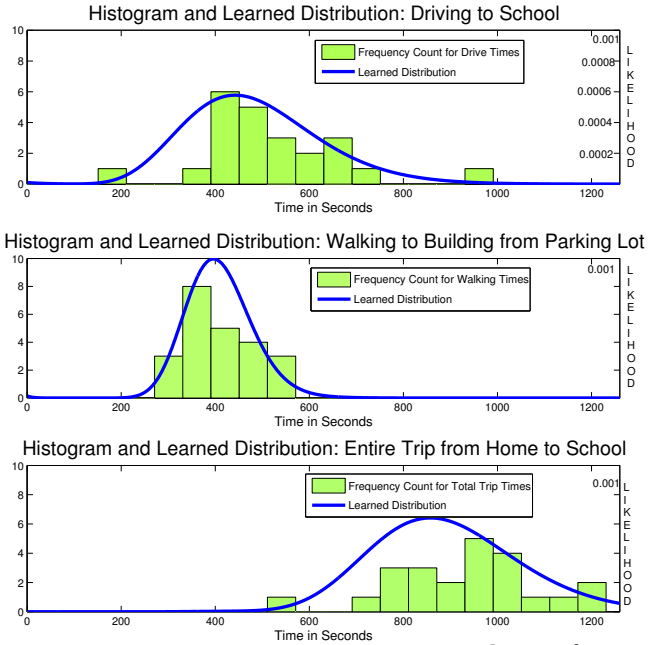

Figure 5: EC-CTBN approximations to travel times for two trip segments and total time to goal.

a city by foot, car, and bus. Specifically, we used data from (Liao, Fox, & Kautz 2004) in which raw, time-stamped GPS data is clustered to find *locations*, *i.e.*, coordinates with large dwell times. Next, a topological movement model is constructed by defining a graph whose nodes are locations and whose edges are called *trip segments*. At any point the user's *activity* is either her current location or the trip segment on which she is engaged. The raw data is hand annotated so that every time point is hierarchically labeled with a GPS coordinate, an activity, and a goal. There are three goals (home, work, and a friend's house), six locations, and fourteen trip segments. We learn a EC-CTBN with two variables (goal and activity) where activity is conditioned on goal. Figure 5 shows the distributions learned for travel times from home to the parking lot and from there to work. Clearly, an exponential distribution would not capture these distributions.

Discussion and Conclusions

Continuous time Bayesian networks are a promising new framework for modeling dynamic processes without committing to a fixed temporal grain size. We have shown how to extend CTBNs to approximately model non-exponential time distributions using Erlang-Coxian approximations. Our method affords tractable learning, but the problem of censored data leads us to use expectation-maximization initialized by the Kaplan-Meier estimate. We further showed how to handle negative evidence in the form of observations that certain variables did not change value over continuous intervals of time. We demonstrated the effectiveness of these extensions by showing that they improved predictive accuracy in two domains: email response behavior and GPS traces of a person traveling about a city.

In our current and future work we are using EC-CTBNs to create much more complex models of a user's work routines; for example, modeling both low-level activities such as using email or preparing documents, and high-level processes such as preparing for a meeting. These more complex models are hierarchically structured and contain both continuous-time and discrete-time nodes, further extensions to the CTBN model we will describe in future papers.

Acknowledgements

This work was supported by NSF grant IIS-0307906, ONR grant N00014-02-1-0932, and DARPA via SRI grant 03-000225. Thanks to Pedro Domingos and anonymous reviewers for insightful comments.

References

Asmussen, S.; Nerman, O.; and Olsson, M. 1996. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics* 23.

Crowder, M.; Kimber, A.; Smith, R.; and Sweeting, T. 1994. *Statistical Analysis of Reliability Data*. Chapman & Hall/CRC.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5:142–150.

Liao, L.; Fox, D.; and Kautz, H. 2004. Learning and inferring transportation routines. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*. Best Paper Award.

Neuts, M. F. 1981. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover.

Nodelman, U.; Shelton, C.; and Koller, D. 2002. Continuous time bayesian networks. In *Proceedings of the Eighteenth International Conference on Uncertainty in Artificial Intelligence.*, 378–387.

Nodelman, U.; Shelton, C.; and Koller, D. 2003. Learning continuous time bayesian networks. In *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence.*, 451–458.

Osogami, T., and Harchol-Balter, M. 2003a. A closed-form solution for mapping general distributions to minimal PH distributions. In *Computer Performance Evaluation / TOOLS*, 200–217.

Osogami, T., and Harchol-Balter, M. 2003b. Necessary and sufficient conditions for representing general distributions by coxians. In *Computer Performance Evaluation / TOOLS*, 182–199.