Learning Parameters of Hybrid Time Bayesian Networks

Manxia Liu LIUMANXIA@CS.RU.NL
Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
Arjen Hommersom ARJENH@CS.RU.NL
Faculty of Management, Science, and Technology, Open University of the Netherlands
Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
Maarten van der Heijden M.VANDERHEIJDEN@CS.RU.NL
Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
Peter J.F. Lucas PETERL@CS.RU.NL
Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands

Abstract

Time granularity is an important factor in characterizing dynamical systems. Hybrid time Bayesian networks model the dynamics of systems that contain both irregularly-timed variables and variables whose evolution is naturally described by discrete time. The former observations are modeled as variables in continuous-time manner and the latter are modeled by discrete-time random variables. We address the problem of learning parameters of hybrid time models from complete data where all the states are known at any time point, and from incomplete trajectories, where continuous-time variables are observed only at some time points. We show that for the complete case, the parameters can be estimated straightforwardly. When some continuous-time variables are (partially) unobserved, it becomes infeasible to learn the parameters in closed form. In that case, we propose to use Markov chain Monte Carlo sampling to estimate the posterior distribution over the parameters. We tested the approach on a number of hybrid time models where continuous-time variables are completely or partially observed, showing that close estimation of the original parameters can be recovered. A medical example is used to illustrate the learning parameters of hybrid time Bayesian networks.

Keywords: Hybrid time Bayesian networks; parameter estimation; missing values; MCMC.

1. Introduction

Many applications involve reasoning about a complex system that evolves over time. Standard frameworks, such as dynamic Bayesian networks (DBNs) (Murphy, 2002), discretize time at fixed intervals and then model the system as evolving discretely from one time slice to the next. Alternatively, in continuous time Bayesian networks (CTBNs) (Nodelman et al., 2002a) states can change continuously over time. In DBNs, the assumption is usually made that the distribution of variables at a particular time point is conditional only on the state of the system at the previous time point. A problem occurs if temporal processes of a system are best described using different rates of change, e.g., one part of the process changes much faster than another. In CTBNs all probabilistic knowledge, for example derived from expert knowledge, has to be mapped to transition rates which are hard to interpret. A new formalism called *hybrid time Bayesian networks* (HTBNs) (Liu et al., 2016) increases the modeling and analysis capabilities for dynamic systems by allowing both irregularly-timed random variables and random variables whose evolution is naturally described by discrete time.

An important task for using any probabilistic model is learning its parameters from data. In most practical cases, we also have to deal with missing values. Although some variables can be measured with a certain fixed frequency, such as daily measurement of drug intake, one cannot observe all variables continuously in many real-world applications. As a consequence of the limitations of the data collection process, some variables may then be observed very irregularly and be missing over the remainder of the time.

In this paper, we describe the learning of the parameters of a hybrid time Bayesian network. In particular we propose to use a Markov chain Monte Carlo (MCMC) method to estimate the posterior distribution of HTBN parameters given partial trajectories. In the next section, we give a brief introduction to HTBNs and introduce a medical example, followed by related work in the literature about learning parameters from data with missing values. Then, in Section 4, we discuss the necessary theory for learning parameters in HTBNs. Subsequently we show the results of experimental tests of the learning approach on data generated from a number of HTBNs. This illustrates how our learning framework can be used to learn parameters from complete and incomplete data. The paper is then concluded with a discussion.

2. Hybrid Time Bayesian Networks (HTBNs)

In this section, we review the *hybrid time Bayesian networks* (HTBNs) framework presented in Liu et al. (2016). An HTBN represents a heterogeneous dynamical system over a finite state space with different evolution rates.

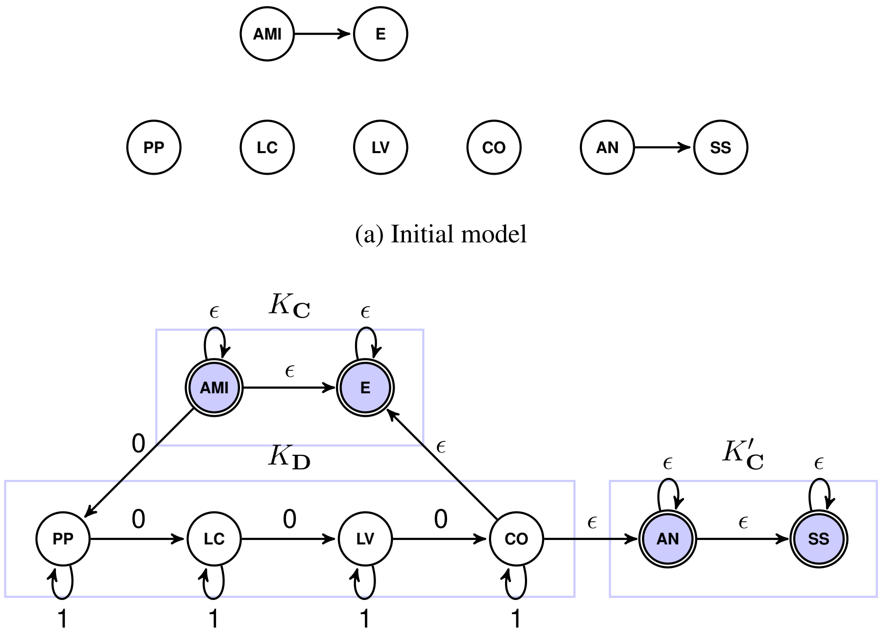
Definition 1 (Hybrid Time Bayesian Networks) *A hybrid time Bayesian network is a tuple $\mathcal{H} = (\mathcal{B}, G_{\mathcal{H}}, \Phi, \Lambda)$, where $\mathcal{B} = (G_0, P)$ is a Bayesian network specifying an initial distribution, $G_{\mathcal{H}} = (V(G_{\mathcal{H}}), E(G_{\mathcal{H}}))$ is a directed graph specifying a transition model with each vertex in $V(G_{\mathcal{H}})$, either a continuous-time variable, collectively denoted by $\mathbf{C}$, or a discrete-time variable, collectively denoted by $\mathbf{D}$, Φ is a set of conditional probability distributions for variables $\mathbf{D}$, and Λ is a set of conditional intensity matrices for variables $\mathbf{C}$.*

Example 1 *Consider the clinical condition of heart failure (Gatti et al., 2012). Heart failure is said to be cardiogenic when the cardiac muscle is the organ from which the circulatory failure is triggered. The strength of the heart muscle is represented by its pump (PP). Cardiogenic heart failure may be caused by acute myocardial infarction (AMI). An AMI reduces blood flow through the coronary arteries to the heart muscle (CO). It manifests as an intense chest pain, called angina pectoris (AN). One consequence is that part of the heart muscle will die, which is revealed later in a blood sample analysis in the lab by an increased level of particular heart muscle proteins, in particular cardiac enzymes (E). Representing this scenario as an HTBN, we have to decide which variables are regular and which are irregular. For example, we model the pump of the heart (PP) as a discrete-time variable, which can be measured regularly, such as on a daily base. On the other hand, the observations of acute myocardial infarction and measurements of cardiac enzymes are usually irregular. In Fig. 1 the complete hybrid time Bayesian network $\mathcal{H}$ is shown.*

Let a *component* be defined as the set of vertices that can be reached by a path, from one vertex to another, ignoring the orientation of the edges. An HTBN can be partitioned into a set of components, where the components consisting of only continuous-time variables are collectively referred to as $\mathbf{K_C}$, and those consisting of only discrete-time variables as $\mathbf{K_D}$; $K_\mathbf{C} \in \mathbf{K_C}$ and $K_\mathbf{D} \in \mathbf{K_D}$ denote individual components. The parents of $V(K_\mathbf{C})$, denoted as $\pi(V(K_\mathbf{C}))$, are a subset $\mathbf{X} \subseteq V(\mathbf{K_D})$ if and only if for every $D \in \mathbf{X}$ there exists at least one arc $(D, C) \in E(G_\mathcal{H})$ with $C \in V(K_\mathbf{C})$. Parents $\pi(V(K_\mathbf{D}))$ are defined analogously with discrete and continuous time variables interchanged. In Fig. 1b, there is one discrete component $K_\mathbf{D}$ over variables PP, LC, LV, CO and parents $\pi(V(K_\mathbf{D})) = \{\text{AMI}\}$ and two continuous components, namely $K_\mathbf{C}$ over variables AMI and E and $K'_\mathbf{C}$ over variables AN and SS with parents $\pi(V(K_\mathbf{C})) = \pi(V(K'_\mathbf{C})) = \{\text{CO}\}$.

The joint probability distribution for hybrid time Bayesian networks factorizes over the conditional joint probabilities for the continuous-time and discrete-time components. Given time points of interest $\mathbf{A}$ for the regular variables and $\mathbf{B}$ for the irregular variables with $\mathbf{A} \subseteq \mathbf{B}$, the joint distribution for an HTBN $\mathcal{H}$ is given by:

$$P(V(G)_\mathbf{B}) = \prod_{K_\mathbf{C} \in \mathbf{K_C}} P(V(K_\mathbf{C})_\mathbf{B} \mid \pi(V(K_\mathbf{C}))_\mathbf{A}) \prod_{K_\mathbf{D} \in \mathbf{K_D}} P(V(K_\mathbf{D})_\mathbf{A} \mid \pi(V(K_\mathbf{D}))_\mathbf{A}) \qquad (1)$$



(a) Initial model

(b) Transition model

Fig. 1: Acute myocardial infarction network. Variables shown are AMI = acute myocardial infarction; E = cardiac enzymes; PP = Pump; LC = volume of blood ejected from left heart ventricle; LV = pressure exerted by the blood; CO = reduction of blood flow; AN = paroxysmal attacks of chest pain; SS = sympathetic nervous system activity. In Fig. 1b, continuous-time variables are graphically represented by double-edged shaded circles, and discrete-time variables by solid nodes. An arc with a number d indicates that the dependence manifests with a delay by time d, such as 0 and ϵ in Fig. 1b.

Example 2 *Consider the example from Fig. 1 with time points of interest* $\mathbf{A}$ *and* $\mathbf{B}$*, and joint intensity matrix* Q *for continuous component* $K_\mathbf{C}$ *over continuous-time variables* AMI *and* E*, and* Q' *for*

$K'_{\mathbf{C}}$ over variables AN *and* SS*. The joint distribution of the transition model P is then given by:*

$$
\begin{aligned}
P = \prod_{\alpha \in \mathbf{A} \setminus \{0\}} & P(\mathrm{PP}_{s(\alpha)} \mid \mathrm{PP}_{\alpha}, \mathrm{AMI}_{s(\alpha)}) P(\mathrm{LC}_{s(\alpha)} \mid \mathrm{LC}_{\alpha}, \mathrm{PP}_{s(\alpha)}) P(\mathrm{LV}_{s(\alpha)} \mid \mathrm{LV}_{\alpha}, \mathrm{LC}_{s(\alpha)}) \\
& P(\mathrm{CO}_{s(\alpha)} \mid \mathrm{CO}_{\alpha}, \mathrm{LV}_{s(\alpha)}) \prod_{\beta \in \mathbf{B} \setminus \{\max \mathbf{B}\}} P(\exp(Q_{\mathrm{CO}_a}(s(\beta) - \beta))) P(\exp(Q'_{\mathrm{CO}_a}(s(\beta) - \beta)))
\end{aligned}
$$

where $a = \max\{\alpha \mid \alpha < \beta, \alpha \in \mathbf{A}\}$ and s is the successor function that gives the next element in an ordering.

3. Related Work

There has been some work dealing with parameter estimation of continuous-time and discrete-time variables from data. When it comes to incomplete data, Expectation Maximization (EM) (Dempster et al., 1977) is widely used for parameter estimation by optimizing a likelihood function. Nodelman et al. (2005) use EM to find maximum likelihood parameters in CTBNs, which defines the duration of a variable staying on a state and the probability distribution over the next state. Unlike EM used for DBNs (Ghahramani, 1998), the sufficient statistics in CTBNs are the time a variable stays in a state and the number of transitions from the state.

In theory, an EM approach would be possible for learning parameters from partially observed trajectories in HTBNs, as HTBNs can be roughly regarded as a mixture of DBNs and CTBNs. However, computing expectations in HTBNs is hard because of the available method for inference in HTBNs. The current approach proposed by Liu et al. (2016) suggests to translate an HTBN into an equivalent discrete-time Bayesian network, called a *representative Bayesian network*, in which inference can be performed using standard exact methods. This translation is computationally hard, which is feasible if the translation has to occur only once. However, in the EM procedure this translation has to be done for every iteration, which makes it an impractical approach.

MCMC is widely used for the evaluation of posterior distribution for Bayesian models (Gilks, 2005; Hastings, 1970). Recently, Rao and Teh (2011) described a Markov chain Monte Carlo (MCMC) approach to perform inference in CTBNs. In this paper, we propose an MCMC-based approach for parameter estimation from incomplete data for HTBNs, in particular when the values of some continuous components are observed at only some time points.

4. Parameter Estimation in HTBNs

The parameter estimation of a hybrid-time model requires the estimation of two types of parameters, i.e. CPTs for discrete-time variables and intensity matrices for continuous-time variables. In this section, we discuss the *maximum a posteriori* (MAP) estimates in HTBNs in order to learn parameters. Throughout this section, we make use of existing results on maximum likelihood estimates and MAP in DBNs (Murphy, 2002) and CTBNs (Nodelman et al., 2002b). In the next subsection, we will first discuss the likelihood function for HTBNs, decomposed in those of DBNs and CTBNs.

4.1 Likelihood of complete data

The *likelihood* for a hybrid time Bayesian network $\mathcal{H}$ is the probability of observations given $\mathcal{H}$'s parameters $\boldsymbol{\theta}$. Let $\mathcal{H}$ be partitioned in components $\mathbf{K_C}$, $\mathbf{K_D}$. Given data $\boldsymbol{\mathcal{D}}$, describing a complete trajectory over all random variables, the log-likelihood for $\mathcal{H}$ can be written as:

$$\begin{aligned}\ell_{\mathcal{H}}(\boldsymbol{\theta} : \boldsymbol{\mathcal{D}}) &= \sum_{K_C \in \mathbf{K_C}} \ell_{K_\mathbf{C}}(\boldsymbol{\theta}_{K_C} : \boldsymbol{\mathcal{D}}) + \sum_{K_D \in \mathbf{K_D}} \ell_{K_\mathbf{D}}(\boldsymbol{\theta}_{K_D} : \boldsymbol{\mathcal{D}}) \\ &= \sum_{K_C \in \mathbf{K_C}} \ln P(V(K_C) \mid \pi(V(K_C)) : \boldsymbol{\mathcal{D}}) + \sum_{K_D \in \mathbf{K_D}} \ln P(V(K_D) \mid \pi(V(K_D)) : \boldsymbol{\mathcal{D}})\end{aligned} \tag{2}$$

Here the log-likelihood is decomposed into the log-likelihood of the continuous-time components, indicated by $\ell_{K_\mathbf{C}}(\boldsymbol{\theta}_{K_C} : \boldsymbol{\mathcal{D}})$, and discrete-time components by $\ell_{K_\mathbf{D}}(\boldsymbol{\theta}_{K_\mathbf{D}} : \boldsymbol{\mathcal{D}})$.

The likelihood of each component of an HTBN is similar to the likelihood in CTBNs and DBNs, conditional on parents which are variables from other components.

Likelihood of discrete-time components Given a discrete component $K_\mathbf{D}$ and data $\boldsymbol{\mathcal{D}}$, the log-likelihood function for component $K_\mathbf{D}$ with parameter $\boldsymbol{\theta}_{K_\mathbf{D}}$ is given by:

$$\ell_{K_\mathbf{D}}(\boldsymbol{\theta}_{K_\mathbf{D}} : \boldsymbol{\mathcal{D}}) = \sum_{D \in V(K_\mathbf{D})} (\ln P(D[0] \mid \pi(D[0]) : \boldsymbol{\theta}_{K_\mathbf{D}}) + \sum_{\alpha \in \mathbf{A} \setminus \{0\}} \ln P(D[\alpha] \mid \pi(D[\alpha]) : \boldsymbol{\theta}_{K_\mathbf{D}})) \tag{3}$$

where $D[0]$ and $D[\alpha]$ are assignments to D at time 0 and α in data $\boldsymbol{\mathcal{D}}$ respectively, $\pi(D[0])$ to the values of parents of D in the initial model, and $\pi(D[\alpha])$ to the values of parents of D in the transition model.

When variables are fully observed over time, the computation of the likelihood can be summarized in terms of sufficient statistics. For discrete components, the sufficient statistics are $M[\mathbf{u}, d]$, $M[\mathbf{u}', d]$, where $M[\mathbf{u}, d]$ is the number of times $D[0] = d$, and $\pi(D[0]) = \mathbf{u}$ and $M[\mathbf{u}', d]$ the total number of times $D[\alpha] = d$ and $\pi(D[\alpha]) = \mathbf{u}'$ for all α, $\alpha \in \mathbf{A} \setminus \{0\}$.

For a discrete-time variable D with parents $\mathbf{U}$ in the initial model and $\mathbf{U}'$ in the transition model, we use parameter $\theta_{K_\mathbf{D}}^{d|\mathbf{u}}$ for each combination of $d \in Val(D)$ and $\mathbf{u} \in Val(\mathbf{U})$ in the initial model for component $K_\mathbf{D}$. Parameter $\theta_{K_\mathbf{D}}^{d|\mathbf{u}'}$ is defined analogously in the transition model. We can then rewrite the log-likelihood function in Eq. 3 as follows:

$$\ell_{K_\mathbf{D}}(\boldsymbol{\theta}_{K_\mathbf{D}} : \boldsymbol{\mathcal{D}}) = \sum_{D \in V(K_\mathbf{D})} (\sum_{\mathbf{u}} \sum_{d} M[\mathbf{u}, d] \ln \theta_{K_\mathbf{D}}^{d|\mathbf{u}} + \sum_{\mathbf{u}'} \sum_{d} M[\mathbf{u}', d] \ln \theta_{K_\mathbf{D}}^{d|\mathbf{u}'}) \tag{4}$$

Likelihood of continuous-time components The distribution over transitions of a process in CTBNs is described with two parts: an exponential distribution over *when* the next transition will occur, parameterized by $\mathbf{q}_{K_\mathbf{C}}$, and a multinomial distribution over *where* the state transits, parameterized by $\boldsymbol{\theta}_{K_\mathbf{C}}$, that is, the next state of the system. Parameters $\boldsymbol{\theta}_{K_\mathbf{C}}$ describe the distribution of states once variables $V(K_\mathbf{C})$ transit, formally, $\boldsymbol{\theta}_{K_\mathbf{C}} = \{\theta_{K_\mathbf{C}}^{cc'} : c \neq c'\}$.

Given a continuous component $K_\mathbf{C}$ with known structure and data $\boldsymbol{\mathcal{D}}$ of $V(K_\mathbf{C})$, the log-likelihood function for component $K_\mathbf{C}$ with parameters $\mathbf{q}_{K_\mathbf{C}}$, $\boldsymbol{\theta}_{K_\mathbf{C}}$ is given by Nodelman et al. (2002b):

$$\ell_{K_\mathbf{C}}(\mathbf{q}_{K_\mathbf{C}}, \boldsymbol{\theta}_{K_\mathbf{C}} : \boldsymbol{\mathcal{D}}) = \ell_{K_\mathbf{C}}(\mathbf{q}_{K_\mathbf{C}} : \boldsymbol{\mathcal{D}}) + \ell_{K_\mathbf{C}}(\boldsymbol{\theta}_{K_\mathbf{C}} : \boldsymbol{\mathcal{D}}) \tag{5}$$

For simplicity, we regard a continuous component as a single continuous-time variable, for which the intensity matrix is computed by the *amalgamation* operation over the variables in the component, as defined in Nodelman et al. (2002a). Thus, the sufficient statistics for a continuous component $K_\mathbf{C}$ can be calculated in terms of $T[c|\mathbf{u}]$, the amount of time $V(K_\mathbf{C})$ spends in state c where $\mathbf{U} = \mathbf{u}$, and $M[c, c']$, the number of times $V(K_\mathbf{C})$ transit from c to c' for $c = c'$. The log-likelihood for continuous component $K_\mathbf{C}$ in Eq. 5 can be computed by summing out the likelihood for all transitions and states, formally:

$$\begin{aligned}\ell_{K_\mathbf{C}}(\mathbf{q}_{K_\mathbf{C}}, \boldsymbol{\theta}_{K_\mathbf{C}} : \boldsymbol{\mathcal{D}}) = &\sum_{\mathbf{u}} \sum_{c} M[c \mid \mathbf{u}] \ln q_{K_\mathbf{C}}^{c|\mathbf{u}} - q_{K_\mathbf{C}}^{c|\mathbf{u}} \cdot T[c \mid \mathbf{u}] \\ &+ \sum_{\mathbf{u}} \sum_{c} \sum_{c' \neq c} M[c, c^{'} \mid \mathbf{u}] \ln \theta_{K_\mathbf{C}}^{cc^{'}|\mathbf{u}}\end{aligned} \tag{6}$$

4.2 MAP estimates with complete data

The parameters that we are interested in are described by the most likely parameters given the data $\boldsymbol{\mathcal{D}}$, formally:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(\boldsymbol{\theta} \mid \boldsymbol{\mathcal{D}}) \tag{7}$$

As usual, for computational efficiency, we use a conjugate prior—one where likelihood is in the same parametric family as the prior. For a discrete-time variable D with a multinomial distribution parameterized by $\boldsymbol{\theta}_{K_\mathbf{D}}$, an appropriate conjugate prior is a Dirichlet distribution, where

$$\boldsymbol{\theta}_{K_\mathbf{D}} \sim \text{Dir}(\gamma_1, \ldots, \gamma_k)$$

After conditioning on the data, the posterior is obtained by adding the prior hyperparameters to the empirical counts:

$$\boldsymbol{\theta}_{K_\mathbf{D}} \mid \boldsymbol{\mathcal{D}} \sim \text{Dir}(\gamma_1 + M[d_1 \mid \boldsymbol{u}], \ldots, \gamma_k + M[d_k \mid \boldsymbol{u}])$$

Discrete-time variables can have different parents in the initial model and transition model; assigning a Dirichlet prior with parameters $\gamma_1, \ldots, \gamma_k$ and $\gamma'_1, \ldots, \gamma'_k$ respectively, we obtain the MAP estimates:

$$\hat{\theta}_{K_D}^{d|\mathbf{u}} = \frac{\gamma_{d|\mathbf{u}} + M[\mathbf{u}, d]}{\gamma_\mathbf{u} + M[\mathbf{u}]} \qquad \hat{\theta}_{K_D}^{d|\mathbf{u'}} = \frac{\gamma_{d|\mathbf{u'}} + M[\mathbf{u'}, d]}{\gamma_\mathbf{u'} + M[\mathbf{u'}]} \tag{8}$$

where

$$\gamma_\mathbf{u} = \sum_d \gamma_{d|\mathbf{u}} \qquad \gamma_\mathbf{u'} = \sum_d \gamma_{d|\mathbf{u'}}$$

Similarly, the conjugate prior in component $K_\mathbf{C}$ for multinomial parameters $\boldsymbol{\theta}_{K_\mathbf{C}}$ is the Dirichlet distribution $\boldsymbol{\theta}_{K_\mathbf{C}} \sim \text{Dir}(\mu^{cc_1|\mathbf{u}}, \ldots, \mu^{cc_n|\mathbf{u}})$. For the exponential parameter $q_{K_\mathbf{C}}$ the conjugate prior is the Gamma distribution, $q_{K_\mathbf{C}} \sim \Gamma(\mu^\mathbf{u}, \eta^\mathbf{u})$. We can then derive the MAP for a continuous component:

$$\hat{q}_{K_\mathbf{C}}^{c|\mathbf{u}} = \frac{\mu^{c|\mathbf{u}} + M[c \mid \mathbf{u}]}{\eta^{c|\mathbf{u}} + T[c \mid \mathbf{u}]} \qquad \hat{\theta}_{K_\mathbf{C}}^{cc'|\mathbf{u}} = \frac{\mu^{cc'|\mathbf{u}} + M[c, c' \mid \mathbf{u}]}{\mu^{c|\mathbf{u}} + M[c \mid \mathbf{u}]} \tag{9}$$

where

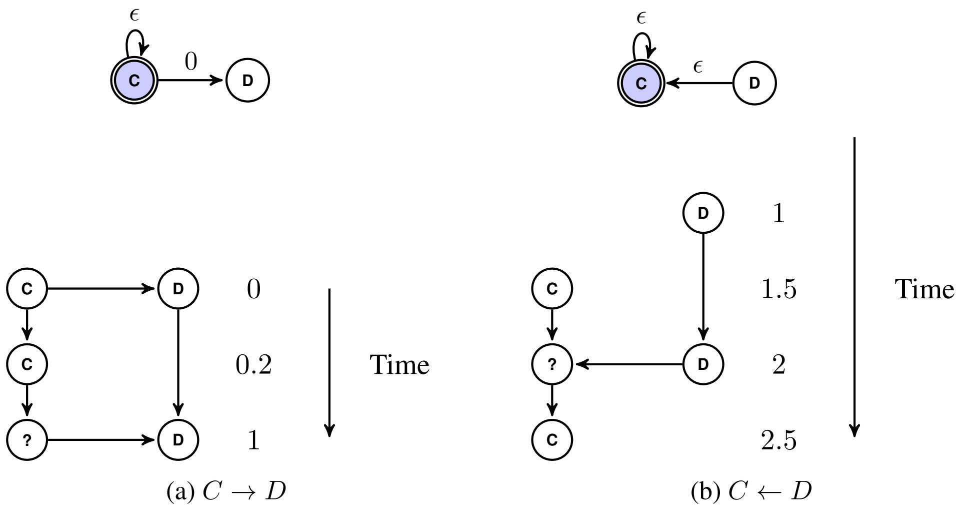
$$\mu^\mathbf{u} = \sum_{c'} \mu^{cc'|\mathbf{u}}$$

4.3 Incomplete data

The likelihood of HTBNs given a partial trajectory is the joint distribution over the observations by summing out the variables which are missing in the data $\boldsymbol{\mathcal{D}}$. Given time points $\mathbf{A}$, $\mathbf{B}$ for an HTBN $\mathcal{H}$ with graph G, parameter $\boldsymbol{\theta}$, and its associated joint distribution $P(V(G)_{\mathbf{B}})$ as defined in Eq. 1, we can compute the log-likelihood of $\mathcal{H}$ given a partial trajectory $\boldsymbol{\mathcal{D}}$ over $\mathbf{A}$ as follows:

$$\ell_{\mathcal{H}}(\boldsymbol{\theta} : \boldsymbol{\mathcal{D}}) = \ln \sum_{V(G)_{\mathbf{B}} \setminus \boldsymbol{\mathcal{D}}} P(V(G)_{\mathbf{B}}) \tag{10}$$

In the following, we are assuming the typical situation where continuous-time random variables are observed at arbitrary points in time, which means that almost everywhere they are unobserved. Since other variables are typically directly related to the value of continuous-time random variables on discrete-time points (by the factorization of an HTBN), this means that we often need to marginalize over the continuous-time variables at those time points. In this general situation, it is infeasible to compute the likelihood in closed form. This is illustrated in the following examples.



(a) $C \rightarrow D$ (b) $C \leftarrow D$

Fig. 2: Two possible structures of HTBNs where continuous components are partially observed.

Example 3 *Suppose we have an HTBN with structure of a discrete component with parents from a continuous component parameterized by* $\boldsymbol{\theta}$ *as illustrated in Fig. 2a. To compute the likelihood of D at time* 1 *given its value at time* 0 *and value of its parent* C *at time* 0.2 *and* C *is unobserved at time* 1*, we need to marginalize out the value of* C *at* 1*:*

$$\begin{aligned}\ell_{\mathcal{H}}(\boldsymbol{\theta} : d_1, d_0, c_{0.2}, c_0) = &\ln P(c_0 : \boldsymbol{\theta}) + \ln P(c_{0.2}|c_0 : \boldsymbol{\theta}) + \ln P(d_0 : \boldsymbol{\theta}) + \\ &\ln \sum_{C_1} P(d_1|d_0, C_1 : \boldsymbol{\theta}) P(C_1|c_{0.2} : \boldsymbol{\theta})\end{aligned}$$

Similarly, suppose the HTBN with the structure of a continuous component which is dependent on a discrete component, as shown in Fig. 2b. The value of C at time 2.5 *is dependent on* D *at time* 2*:*

$$\begin{aligned}\ell_{\mathcal{H}}(\boldsymbol{\theta} : c_{2.5}, d_2, c_{1.5}, d_1) =& \ln P(d_1 : \boldsymbol{\theta}) + \ln P(d_2|d_1 : \boldsymbol{\theta}) + \ln P(c_{1.5} : \boldsymbol{\theta}) + \\ & \ln \sum_{C_2} P(c_{2.5}|d_2, C_2 : \boldsymbol{\theta}) P(C_2|c_{1.5}, d_1 : \boldsymbol{\theta})\end{aligned}$$

In order to obtain MAP estimates in this case, we resort to MCMC sampling to maximize the posteriors. Results are discussed in the next section.

5. Experiments

In this section, we experimentally explore learning parameters of HTBNs from complete and incomplete trajectories. First an algorithm is presented that generates complete trajectories of HTBNs, including all states and transition time points. Then, the trajectories are used to generate partial trajectories by sampling at random time points. Subsequently, we use the mentioned complete and incomplete trajectories to learn the parameters for a number of HTBNs. We then evaluated the learned model in terms of log-likelihood on new complete trajectories.

5.1 Data generation process for HTBNs

In this section, we describe the procedure for generating complete trajectories from a parameterized HTBN, i.e., all the states of variables and associated time points where a transition occurs. The generated trajectories are later used for parameter estimation of HTBNs and model evaluation in Section 5.3.

The data generation procedure can be seen as generating a sequence of *event*, where an event is a pair $\langle X \leftarrow x, \tau \rangle$, which indicates a variable X that either evolves continuously over time or is observed regularly, takes value x at time τ. Let α and $s(\alpha)$ be the current and next system time for all discrete components, $\alpha, s(\alpha) \in \mathbb{N}_0$. Let β_C be the current system time for a continuous component K_C. Each continuous component is described by its own system time, which results from the fact that continuous component states can evolve at different rates.

Let $\mathbf{K_C}$, $\mathbf{K_D}$ be a partition of an HTBN $\mathcal{H}$. We initialize σ as an empty event sequence. The parents of a continuous component are denoted as $\pi(V(K_C)))$, with $\pi(V(K_C))) \subseteq V(\mathbf{K_D})$. A set of variables $\pi(D^i)$ refer to the parents of a discrete-time variable D^i. Initially, we have $\alpha = 0$, $\beta_C = 0$, and the states of components are initialized by sampling at random from the initial BN. Let c^i and d^i be the starting states of components K_C, K_D, $K_C \in \mathbf{K_C}$, $K_D \in \mathbf{K_D}$. We can generate the event sequence σ by repeating the steps in Fig. 3 with the following order: Fig. 3a → Fig. 3b → Fig. 3c → Fig. 3a, where each loop selects events for continuous components to occur between two successive time slices α and $s(\alpha)$, and events for discrete components at time $s(\alpha)$.

The states of components are propagated between different types of components, as shown in Fig. 3. Firstly, the current states of discrete components at α are propagated to their corresponding continuous components. We then choose the intensity matrix $Q_{K_C|\pi(V(K_C))_a}$, $a = \max\{\alpha \mid \alpha \leq \beta_{K_C}\}$, for a continuous component K_C with the current configuration of its parents $\pi(V(K_C))$ at time a. Once we have intensity matrices of continuous components, we search for all events of continuous component that take place between time α and $s(\alpha)$, as shown from line 4 to 6 in Fig. 3a and Fig. 3b. It is followed by propagating the states of continuous components at $s(\alpha)$ to discrete components in order to sample discrete variables at time $s(\alpha)$, as shown in Fig. 3c.

```
while $s(\alpha) \leq N$ do
    foreach $K_C \in \mathbf{K}_C$ do
        Let intensity matrix for $K_C$ be $Q_{K_C|\pi(V(K_C))_a}$, $a = \max\{\alpha \mid \alpha \leq \beta_{K_C}\}$
        while $\beta_{K_C} < s(\alpha)$ do
            Let $q^i$, $q^{ij}$ be intensity associated with its current state $c^i$
            goto 3b
    foreach $K_D \in \mathbf{K_D}$ do
        goto 3c
    $\alpha = s(\alpha)$
```

(a) **Algorithm 1** Data generation

```
$\tau \sim \text{Exp}\left(q^i\right)$
if $\tau + \beta_{K_C} \leq s(\alpha)$ then
    $\beta_{K_C} = \beta_{K_C} + \tau$
    Choose state $V(K_C) \leftarrow c^j$ with probability $q^{ij}/q^i$
    Add event $\langle V(K_C) \leftarrow c^j, \beta_{K_C} \rangle$ to $\sigma$
else
    $\beta_{K_C} = s(\alpha)$
```

(b) Generate next continuous states

```
Let $D^1, \cdots, D^n$ be a topological order of $V(K_D)$
for $i \in 1 : n$ do
    Sample $d^i$ from $P(D^i_\alpha \mid \pi(D^i_\alpha))$, where $\pi(D^i) \subseteq V(\mathbf{K_C}) \cup V(\mathbf{K_D})$
    Add event $\langle D^i \leftarrow d^i, s(\alpha) \rangle$ to $\sigma$
```

(c) Generate next discrete states

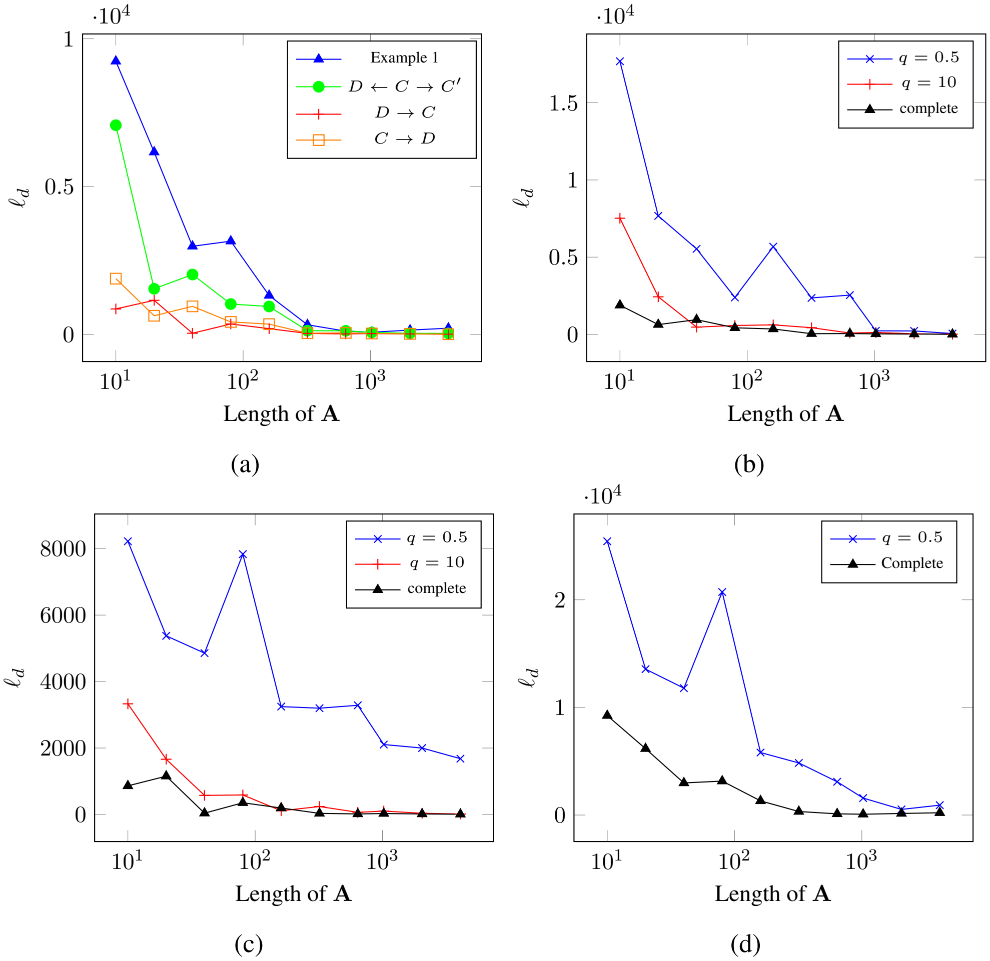
Fig. 3: Data generation procedure for HTBNs. 3b: sample an event for a continuous component between α and $s(\alpha)$; 3c: sample events for discrete components at time $s(\alpha)$.

5.2 Experimental setup

For learning from complete data, parameters of HTBNs were learned using the exact MAP estimates as discussed in the previous section. The MCMC sampling approach for partial trajectories was implemented in *RStan*[1], an R-interface to the *Stan* probabilistic programming language[2]. We set the number of total iterations to 1000, including the *burn-in* stage. We drew the multinomial parameters of the network from Dirichlet distributions with parameters all equal to 1, and the exponential parameters from a Gamma distribution with both parameters set to 2. We tested the learning performance to learn parameters from complete and partial trajectories with different length in terms of the number of discrete-time slices $\mathbf{A}$.

[1] Stan Development Team. 2016. RStan: the R interface to Stan, Version 2.9.0. `http://mc-stan.org`

[2] The model description and experiments with more details: `http://www.cs.ru.nl/M.Liu/publication.html`



(a)

(b)

(c)

(d)

Fig. 4: Log-likelihood distance ℓ_d for various HTBNs learned from data. (a) Learning with complete trajectories with structure $D \rightarrow C$, $C \rightarrow D$, $D \leftarrow C \rightarrow C'$, and the model in Example 1. Figures $(b - d)$ compare learning with complete and partial trajectories for three structures, with (b) $C \rightarrow D$, (c) $D \rightarrow C$, (d) the HTBN as shown in Example 1.

We tested on various synthetic data sets, generated from HTBNs according to the procedure defined in Section 5.1. There are a number of methods to quantify the quality of learned models, such as the KL-divergence. As computing this measure is computationally hard for larger models, we evaluate on a testset generated from the true model. In the following, we fix a large testset, and evaluate the quality of the model in terms of the distance between the log-likelihood of the data on the original model vs the learned model. Formally:

$$\ell_d = |\ell_t - \ell_e| \tag{11}$$

where ℓ_t is the log-likelihood given the true model, and ℓ_e given the learned model.

5.3 Results

We first evaluated the approach to learn from complete trajectories, see Fig. 4a. The learning of HTBNs converges very quickly to the true parameters for HTBNs that contain only a single continuous and discrete-time variable. Similar results are obtained for larger models, convergence is obviously slower since these models contain more parameters.

We further tested our ability to learn parameters of HTBNs from partial trajectories considering missing values for continuous-time variables. We first sample complete trajectories for continuous-time variables including all the states and time points when the transitions occur. We then randomly generated time points from an exponential distribution with rate q, to construct point-based evidence, i.e., partial trajectory, where we know nothing between two time slices. The rate q is associated with the length of the sequence, i.e., the higher the rate is, the more observations we have on the partial trajectory. We tested our ability to learn parameters from partial trajectories with several rates q. As we can see in Fig. 4b and 4c, the estimation of parameters is improved by giving a larger sequence that is generated by a higher rate. It also suggests we can recover the true parameters from partial trajectories that are generated from a sufficiently high rate. Finally, we tested the learning performance on a more complicated hybrid model as shown in Example 1, where there is a cycle between continuous and discrete component. To evaluate if we have similar convergence of the model given a small dataset, we tested the approach to learn parameters from partial trajectories generated from a smaller rate. As you can see in Fig. 4d, also in this case, models of high quality can be learned using the sampling approach.

6. Conclusion

In this work, we addressed the problem of parameter estimation of HTBNs from complete and partial trajectories. For continuous-time variables that are only observed at some time points, we proposed to use MCMC to estimate the posterior distribution over the parameters. This learning approach was tested on various HTBNs with different structures and complexities. The experiment shows that we can get a close estimation of the distribution of HTBNs from partial trajectories. Besides, the experiments also suggest that partial trajectories sampled at a higher rate increase the learning rate.

There are several aspects that will be interesting to explore in the future. Firstly, it will be interesting to see whether the estimated parameters of discrete-time and continuous-time variables converge to the true parameters at same pace. We expect that parameters related to the continuous-time variables are harder to estimate because the exact time-point where transitions occur is unknown in partial trajectories. Secondly, at the moment we focused on a comparison of learning models with different complexities in terms of number of variables and parameters. However, the additional complexity of HTBNs is primarily in the dependence between discrete-time and continuous-time variables. We expect that in real-world applications there will a significant amount of such dependences. Therefore, in a follow-up we will also investigate the quality of learned models for HTBNs with a varying number of components.

Acknowledgments

Manxia Liu is supported by China Scholarship Council.

References

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

E. Gatti, D. Luciani, and F. Stella. A continuous time Bayesian network model for cardiogenic heart failure. *Flexible Services and Manufacturing Journal*, 24(4):496–515, 2012. ISSN 1936-6582. doi: 10.1007/s10696-011-9131-2.

Z. Ghahramani. Learning dynamic Bayesian networks. In *Adaptive processing of sequences and data structures*, pages 168–197. Springer, 1998.

W. R. Gilks. *Markov chain Monte Carlo*. Wiley Online Library, 2005.

W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

M. Liu, A. Hommersom, M. van der Heijden, and P. Lucas. Hybrid time Bayesian networks. *International Journal of Approximate Reasoning*, 2016.

K. P. Murphy. *Dynamic Bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

U. Nodelman, C. R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 378–387. Morgan Kaufmann Publishers Inc., 2002a.

U. Nodelman, C. R. Shelton, and D. Koller. Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 451–458. Morgan Kaufmann Publishers Inc., 2002b.

U. Nodelman, C. R. Shelton, and D. Koller. Expectation maximization and complex duration distributions for continuous time Bayesian networks. In *Proceedings of the Twenty-First Conference on Uncertainty in Artifical Intelligence*, pages 421–430. AUAI Press, 2005.

V. A. Rao and Y. W. Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the 27th International Conference on Uncertainty in Artificial Intelligence*. 2011.