Time and Bayesian Networks

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.H.J.M. van Krieken
volgens besluit van het college van decanen
in het openbaar te verdedigen op donderdag 30 augustus 2018
om 12.30 uur precies

door

Manxia Liu
geboren op 5 oktober 1987
te Zhejiang (China)

Promotor: Prof. dr. P.J.F. Lucas

Copromotoren: Dr. A.J. Hommersom (OU)
Dr. M. van der Heijden (FocusCura, Amsterdam)

Manuscriptcommissie:
Prof. dr. ir. M.J. Plasmeijer
Prof. dr. M.G. Madden (National University of Ireland Galway)
Dr. A. Tucker (Brunel University London, United Kingdom)

SIKS Dissertation Series No. 2018-20

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems, the Institute for Computing and Information Sciences of the Radboud University, the COPD+ project supported by EFRO, with financial support by the China Scholarship Council, and by a grant from project NanoSTIMA [NORTE-01-0145-FEDER-000016], which was financed by the North Portugal Regional Operational Programme [NORTE 2020], under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund [ERDF].

Cover picture: Dreamstime/Alexmax
Printed by: Gildeprint
ISBN: 9789492896490

CONTENTS

1

INTRODUCTION

The physical world we live in is characterized by evolution and ongoing changing processes. While this is true for almost anything in the world, it is in particular true for the main application field of the research in this thesis: the field of clinical medicine. For example chronic diseases are usually not fully stable but, by influencing the underlying pathophysiological mechanisms, their associated signs and symptoms are affected by changes in the patient's environment, the interaction with other diseases that temporally may coexist, and of course, by various treatments. In addition, a disease evolves in time, and when it concerns a *chronic* disease, will often become more severe as time progresses. Developing models that accurately describe disease evolution helps us to better understand in which way the human body reacts to the disease and whether there are suitable ways to recover from it. It also helps in designing new, more effective treatments.

This thesis is about modeling evolving systems using probabilistic graphical models as main representation technique, with its linked methods of probabilistic reasoning and learning. Here we restrict ourselves to dynamic systems, where different parts of the systems may evolve at their own rate, and where the time for the system to change from one state to another is also an important aspect of modeling the dynamics.

1.1 COMPLEX SYSTEMS

Many real-world systems are composed of a large number of parts that interact with each other; these systems are known as *complex systems*. As the name indicates, modeling the behavior of the systems can be intrinsically difficult due to the complicated interactions between their parts. This is indeed also a typical situation in the medical field. For many medical problems, we may be interested in a patient's health status in terms of the presence of a particular disease or illness. A more

concrete example is modeling a patient's heart condition. There are a number of factors that are associated with the patient's cardiac health status, such as the presence of angina pectoris, in most cases indicating the presence of coronary artery disease. In the end, coronary artery disease may progress towards myocardial infarction, commonly called a "heart attack". A heart attack usually results in damaged heart muscle, that is partly replaced by non-functional scar tissue. The reduction in functional heart muscle is a major contributor to heart failure, i.e., the heart is no longer able to comply with the body's circulatory demands. There are drugs available that counteract heart failure, such as digitalis. It is a drug that is also known because of its narrow therapeutic spectrum, making it one of the favorite toxic substances by the murderers in Agatha Christie's detective novels. To build a system that appropriately describes a patient's heart condition, unarguably these factors must be taken into consideration and the resulting system, therefore, will involve complicated interactions between these factors.

One major characteristic of complex systems we in particular pay attention to here is *uncertainty*. It is nearly impossible to precisely know what the future state of a system will be. Uncertainty can be due to the unavailability of complete information about a system, or because the behavior of a system is not well understood [74]. The causes of uncertainty in medicine can be more subjective and complicated. In the case where patients are involved to self evaluate the state of their illness, patients have also been shown to experience "uncertainty in illness" [62], suggesting that patients are incapable of determining the meaning of illness-related events. A more fundamental problem is that a shared concept of uncertainty from different disciplines is still missing [88].

Dealing with uncertainty is also a major issue in the primary domain of this thesis, viz., medicine, where the object of interest is an extremely complex biological machine: the human body. It is clear that, without incorporating a manner of dealing with uncertainty in medical software systems, one can not build systems that are capable of handling uncertain events in actual clinical circumstances.

1.2 THE DYNAMICS OF COMPLEX SYSTEMS

Not only do systems consist of many complex interacting parts, they also manifest rich *dynamic behavior*: most parts of systems are constantly changing and evolving over time. The effect of one part of such a system

on another may not take place immediately, but instead only after a certain amount of time.

Capturing the dynamics of complex systems in models has many potential benefits, and we are in particular interested in its use in predicting the course of a disease. In the case of heart failure, a relevant predictive question of clinical interest would be how likely it is that a patient's heart condition is stabilized in the following days after the patient has had a heart attack, followed by the administration of medical treatment for a certain amount of time. To be able to answer such a question, we are not only concerned with the static state of the patient's heart condition, but also with *when* the patient reaches a desired health state. Having the answers to these questions can be used to assist a physician's clinical judgment in the process of disease management. For example, during the process of making a diagnosis, the temporal order of presence and absence of signs and symptoms related to heart failure can be used for a more timely intervention and more effective treatment.

An important aspect of describing the dynamics of complex systems is *time granularity*, i.e., the level of temporal abstraction at which information is expressed. The changes in parts of dynamic systems can happen at different time granularities and on different time scales. In a model describing heart failure, for example, a drug may be administered frequently, such as on a daily basis, whereas the occurrence of a heart attack can be far less frequent with an average rate of once every 7 years. In addition to varying evolving rates, a part in the system may change irregularly over time. Instead of the part changing at a constant rate, the time for a change can be random, but may follow a probability distribution. This can be illustrated by the incubation time, the time for an individual to manifest the symptoms of a viral infection, such as the flu. It is clear that the incubation time varies from person to person, as individuals vary greatly in how they react to pathogens.

To formally describe dynamic complex systems as models, many different techniques are being used. Usually the behavior of these systems is described by employing systems of differential equations, whereas when there is a focus on discrete state descriptions, timed automata are used. We, however, opt for *Bayesian networks* (BNs) in this thesis, a graphical representation of a joint probability distribution, as they are considered to be particularly suited for capturing and reasoning with uncertainty that comes with dynamic complex systems. Probability the-

ory lays the foundations of Bayesian networks and offers a well-founded and mathematically sound basis for representing and reasoning with uncertainty.

1.3 INTELLIGENT SYSTEMS FOR CLINICAL DECISION SUPPORT

Clinical medicine has been one of the first areas in which probabilistic methods were applied to support decision making (e.g. [16, 30, 89]). These early research efforts were mainly motivated by the complexity of medicine and the high likelihood that errors were made by the medical doctors in diagnosing and treating illness. However, the early probabilistic methods were very limited in their capability. As a consequence, some researchers from Artificial Intelligence started to work on methods that were more powerful. Initially the focus was on representing medical knowledge by making use of clinical expert knowledge, giving rise to software systems that were initially called *expert systems* and later *knowledge-based systems* [59].

An expert system is a computer program that contains knowledge of one or more human experts in a specific domain. The ultimate goal of an expert system is to emulate the decision-making ability of a human expert in a specialized area to support decision makers who are less experienced. One essential component in an expert system is the knowledge base [59]. The knowledge base represents facts about the world and a set of if-then rules that relate these facts to each other to allow drawing conclusions. Clearly, having a knowledge base of a given domain describing a problem of interest is a prerequisite for building an expert system. The process of eliciting and formalizing knowledge for building a knowledge base is known as *knowledge acquisition*.

Early on it was recognized that a drawback of knowledge-based systems is the so-called *knowledge-acquisition bottleneck*: acquiring and modeling knowledge from experts is hard, very time-consuming, and it is usually not possible to obtain a description of the problem which covers all the possibilities. To make matters worse, experts are by definition highly-valued and in constant demand by organizations, and their time for knowledge elicitation is very limited. Several attempts were thus made to automate the process of knowledge acquisition to overcome the knowledge-acquisition bottleneck. In the 1980s, much research focused on developing software tools for knowledge acquisition, to help automate the process of designing and maintaining

rules designed by experts. Unfortunately, the success of this research in the end was very limited. The main result of this research has been a knowledge management, analysis and development methodology, called CommonKADS [85]. The methodology is in limited use today, although some ideas from CommonKADS are being employed in computer-based clinical guidelines design and building clinical decision-support systems [43, 95].

In the last decades, the focus has shifted from expert knowledge to medical data for building medical systems. Broadly speaking, medical data contains health-related information, which is associated with regular patient care or as part of a clinical trial program. Medical data takes the forms of, but is not limited to, electronic health records, administrative data, claim data, data of disease registries, health surveys and clinical trial data, etc. These medical data are relatively easier to obtain than trying to extract medical knowledge from experts, thus it has the potential of overcoming the knowledge-acquisition bottleneck.

To make the best use of medical data, a broad class of algorithms have been devised to be able to learn different kinds of models from data that, e.g., can be used for making predictions. Nowadays, the field of *machine learning*, a term coined by Arthur Samuel in 1959 [82], includes a plethora of different algorithms, models and techniques to be able to learn from data. These models sometimes perform as good, or even better, as human experts. It has strong ties to mathematical optimization, which delivers methods, theories and application domains to the field. It is not surprising that models learned from data may also reveal surprising associations between signs and symptoms over time that our human brain would never suspect. Such associations may assist us in better understanding the evolution of a disease, and even how signs and symptoms regarding a disease develop over time.

1.4 DATA COLLECTING TOOLS: PAPER VERSUS MOBILE DEVICES

Collecting medical data is indispensable for applying machine learning to medical problems. Medical data usually concerns a patient's general information, including age, gender, ethnic origin, and disease history. It is augmented with additional information from the medical interview between the patient and physician, results from physical examination, radiology, laboratory results, medication and other treatments, etc. With

the exception of the general information, most of the other information in the medical data is time related, i.e., it is time stamped.

In this thesis, we are mainly concerned with clinical data that are obtained from patients in their home environment. In the past, most of the data analyzed by medical statisticians were obtained from specialized clinical studies or clinical trials, in which different treatments were compared. Nowadays, increasingly medical data are being collected in the home environment. The home environment is the habitat in which the patient having a chronic disease, such as cardiovascular disease or chronic obstructive pulmonary disease (COPD), resides most of the time. Understandably, clinicians have realized that it is crucial to understand how the patients react to treatments under the 'normal' conditions of the home.

In the past, it has been very hard to collect medical data in the home environment, because researchers did not have the ability to control the data collection process. Early attempts have focused on using paper diaries and questionnaires to collect medical data. However, it may be hard to ensure that paper diaries are filled out by patients on a regular, very frequent basis for a long period of time. In addition, some of the questionnaires can be very lengthy. In the case where information during the day is required at multiple time points, there is obviously no guarantee that the respondent will complete a questionnaire according to the protocol when at home. Without a clear guidance on the regularity of entries and follow-up encouragement, there is a danger that a questionnaire will become a one-off task.

Recently, there is a dramatic shift in the medical data collecting tools in the research community, from conventional paper diaries and questionnaires to more advanced technology-based tools, in particular mobile devices. Mobile devices are increasingly common in today's everyday life and impacting profoundly in shaping the way we live. As communication tools, they also facilitate better maintenance of connections between patients and physicians, thus dissolving the boundaries that separate inpatient and outpatient care. This is particularly important for patients with chronic diseases, who are able to control their disease in the comfort of their own homes, as it makes it possible that the patients can receive timely health care of high quality in spite of the physical disconnection between the patients and their physicians. Not surprisingly, researchers in medicine and other related fields are also seeking new

ways of using mobile devices as a data collection tool in clinic studies, where so far traditionally paper diaries have been mainly employed.

Mobile devices are a promising substitute for paper diaries and questionnaires in the process of data gathering by overcoming their limitation of low accuracy. Unlike paper diaries, compliance of patients to research protocols can be better enforced via a user-friendly reminder using mobile devices. The reminder can be sent out automatically by a preset alarm clock in the patients' mobile devices or a phone call from supervising researchers, when it is detected that the entries are not registered by the patients according to the protocol. These automatic reminders help the patients to capture their experiences close to the time of the occurrence of the symptoms and signs related to their diseases and illness, thus improving the accuracy of the gathered data.

In addition to higher accuracy, mobile devices are an effective tool to get access to additional information that can not be easily captured by paper diaries. Mobile devices provide a unique opportunity to capture the way in which individual participant interacts with recording systems, which can be used to iteratively refine and enhance the usability and usefulness of developed systems. Equipped with more advanced functionalities, mobile devices can identify whether participants in clinical trials intentionally disengage themselves from registering entries in time. With such information, it allows us to potentially reduce the number of participants in clinical trials based on their engagement levels. More importantly, more objective and real-time measures of the patient's health condition can be collected. For example, a patient's physical activity behavior can be quantified and be continuously monitored by built-in sensors. This is particularly suitable for the family members of patients requiring constant healthcare while still living a solitary life, to be acknowledged of the patient's current health condition and to be called for timely healthcare intervention.

A recent successful intelligent medical application using mobile devices is the autonomous mobile system for the management of chronic obstructive pulmonary disease (COPD) [42]. Fig. 1.1 shows the set-up of this eHealth application for patient management and home monitoring as also used in the research of this thesis. In particular, the data collected in this way were used in our research.

Despite that the new emerging tools offer many new opportunities, it also yields a new type of medical data. Unlike paper diaries, where a time interval is predefined, patients may have the autonomy to de-

Fig. 1.1: Typical architecture of an eHealth application, in this case for home monitoring of patients with COPD. It consists of sensors and a smartphone app that is used to give the patients feedback based on a probabilistic interpretation of sensor measurements and answers to a questionnaire, and a back-end application in the hospital that stores patient data supporting follow-up by medical specialists.

cide *when* they make a registration, depending on their preference. In practice, patients may receive contrasting instructions of how often a registration should be made. On one hand, a message from a smartphone will remind a patient to make regular registrations, often on a daily basis. On the other hand, they also receive instructions directly from clinical researchers that a registration is made only when they feel something abnormal. Apparently, the latter case is more likely to be seen in clinical data collection since it is the most time-saving process from patients' perspective. As a consequence, the collected data will take a particular form where observations are made at irregularly spaced time points. The time irregularity imposes a new challenge in applying machine learning techniques.

1.5 MODELING TECHNIQUES

Bayesian networks have become the most widely accepted technique for incorporating expert knowledge along with data into one probabilistic model. Expert knowledge can be incorporated into models by either constructing the causal (or independence) graph, or by incorporating

factors into the causal network which are important for inference but cannot be well captured by data. Initially, Bayesian networks were used to build probabilistic expert systems (e.g. Pathfinder, [38, 39]), which were completely based on domain-expert judgements. Later machine learning algorithms were developed to learn Bayesian networks from data (e.g. [12]). As a consequence, it has become possible to use expert knowledge as background knowledge in the Bayesian network learning process, as such is often done nowadays in probabilistic machine learning, e.g. in the R package BNLearn [65].

Widely used and well-known extensions of Bayesian networks are dynamic Bayesian networks (DBNs), supporting the modeling of dynamic probabilistic systems [14, 64]. DBNs extend standard Bayesian networks by assuming that changes in a system can be captured by a sequence of states of the system at discrete time points. Usually the assumption is 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 a system is best described using different rates of change, e.g., one temporal part of the system changes much faster than another. In that case, the whole system has to be represented using the finest time granularity, which is undesirable from a modeling and learning perspective. In particular, if a variable is observed irregularly, much data on discrete-time points will be missing and conditional probabilities will be hard to estimate.

As an alternative to DBNs, dynamic complex systems can also be modeled as continuous time Bayesian networks (CTBNs), where time acts as a continuous parameter [68]. In these models, the time granularity is infinitely small by modeling transition rates rather than conditional probabilities. Thus, multiple time granularities, i.e., slow and fast transition rates, can easily be captured. A limitation from a modeling perspective is that all probabilistic knowledge, for example derived from expert knowledge, has to be mapped to transition rates which are hard to interpret. Moreover, it is assumed that the transition times, the time until a transition occurs, are exponentially distributed, which may not always be appropriate.

1.6 OUTLINE OF THE THESIS

In this thesis, we narrow down our research to temporal Bayesian networks and to medical problems where evolution of a disease is a ma-

jor concern. More specifically, we try to answer the question that what are the advantages and disadvantages of using DBNs and CTBNs to model medical data where observations are made regularly and irregularly spaced in time. Another question we also try to address is how to further improve the existing CTBNs to be better-suited for general real-world problems. Our choice of improving CTBNs is based on the continuous nature of dynamics systems, which is what exactly described in CTBNs.

For the first question, we conduct comparative research of DBNs and CTBNs to deal with both regular and irregular clinical data for a real-world problem. For the second question, we present two extensions of CTBNs by allowing richer and more flexible time distributions and by incorporating both discrete and continuous time in a model. Now we give an overview of the remaining chapters in this thesis and summarize their contributions.

Chapter 2: Preliminaries

In the next chapter, we give a brief overview of some necessary technical preliminaries on probability theory, probabilistic graphical models and Markov chains.

Chapter 3: Hybrid time Bayesian networks

In this chapter, a new type of temporal Bayesian network model, called hybrid time Bayesian networks (HTBNs), is proposed. It is shown how HTBNs can be used to model dynamic systems, and in particular it is shown that HTBNs allow modeling the temporal behavior of variables based on their associated granularity. For variables for which expert knowledge is available to estimate their parameters, it is sufficient to describe them at discrete time points. Whereas for variables with available data that show significant time irregularity, it is better to incorporate time as one of their continuous parameters. In HTBNs, these two types of variables are allowed to coexist. Another contribution of this chapter is the combined representation of DBNs and CTBNs into generalized temporal probabilistic models. The potential benefits of HTBNs are illustrated by an example model of the temporal evolution of heart failure. Furthermore, we establish a mapping of hybrid-time networks into standard BNs given a set of time points of interest. The inference problem in HTBNs is therefore reduced to a problem for which efficient solutions

exist. This chapter is based on the paper titled *"Hybrid time Bayesian networks"* published in the conference ECSQARU, 2015 (best student paper award) [53] and the journal IJAR, 2017 [55].

Chapter 4: Learning parameters of hybrid time Bayesian networks

In this chapter, we continue the research described in Chapter 3 by defining algorithms for parameter estimation of HTBNs from complete as well as incomplete data. We use MAP estimation of parameters when training data is complete. When the training data contains missing values, we use a Markov Chain Monte Carlo (MCMC) method implemented in the widely used probabilistic programming language Stan to estimate the posterior distribution of parameters in HTBNs. We use these approaches to learn a number of HTBNs with different structures and complexities. In the incomplete data case, however, it is assumed that only the variables that are changing continuously in time are allowed to have missing values. This chapter is based on the paper titled *"Learning parameters of hybrid time Bayesian networks"* [54] published in the conference PGM, 2016.

Chapter 5: Modeling unevenly spaced clinical time series using temporal Bayesian networks

In this chapter, we investigate the capability of DBNs and CTBNs to learn from clinical time series that vary in nature. In order to compare the two temporal Bayesian network types for regularly and irregularly spaced time-series data, three typical ways of observing time-series data are investigated: (1) regularly spaced in time with a fixed rate; (2) irregularly spaced and missing completely at random at discrete time points; (3) irregularly spaced and missing at random at discrete time points. In addition, similar experiments are carried out using real-world COPD patient data where observations are unevenly spaced. This chapter is based on the paper titled *"Modeling clinical time series data using temporal Bayesian network"* [58].

Chapter 6: Making continuous time Bayesian networks more flexible

In this chapter, we propose an extension to support the modeling of the transition time with a richer and more flexible distribution, rather than the exponential distribution in the standard CTBNs. To describe such a

distribution, we introduce an additional hidden variable. With the hidden variable, we also allow CTBNs to obtain memory, which is lacking in standard CTBNs. In addition, two learning methods are proposed and compared to estimate parameters for the proposed models. This chapter is based on the paper titled *"Representing hypoexponential distributions in continuous time Bayesian networks"* [56] and *"Making continuous time Bayesian networks more flexible"* [57] published in the conferences IPMU and PGM, 2018, respectively.

Chapter 7: Conclusions and further research

In this chapter, we place the main contributions of this thesis in a more general context and conclude with an outlook on possible future research.

2

PRELIMINARIES

In this chapter, we review some important theoretical background regarding key concepts in probability theory, discrete and continuous time Markov processes, Bayesian networks and their temporal variants, i.e., dynamic Bayesian networks and continuous time Bayesian networks. This material is included in a separate preliminary chapter, since it forms the basis for most of the development in the remainder of the thesis. All of the material is intended to provide a minimal subset of theoretical knowledge to support understanding most of the discussions in the thesis, rather than to offer a comprehensive review of these fields.

2.1 NOTATION

To provide a clear understanding of the technical terms employed in this thesis, we first start with reviewing the notations we will use. We use upper-case letters or strings, e.g., X, Y, to denote random variables and bold upper-case letters e.g., $\mathbf{X}$ and $\mathbf{Y}$, to denote a set of random variables. For a binary variable with values *true* and *false*, its values are also denoted by lower-case letters x, and $\bar{x}$, short for $X = \mathit{true}$ and $X = \mathit{false}$, respectively. Alternatively, we use $X = T$ and $X = F$, and $X = 1$ and $X = 0$, respectively. If the value of a variable is known, this is often referred to as an *observation*, an *instantiation*, or *evidence*.

Often a random variable X is indexed by discrete or continuous time t, which is then indicated by X_t. In addition, we will make use of a successor function s, which is defined on a countable, linearly ordered set of numbers Z in which every element $z_i \in Z$ with index i is mapped to element $s(z_i) = z_{i+1} \in Z$. If the set Z consists of natural numbers, then we also assume that $s(z_i) = z_i + 1 = z_{i+1}$, with $z_i \in Z$.

Finally, in the thesis we will sometimes employ some notions from linear algebra. In more advanced algebraic expressions, vectors $\mathbf{v}$ are assumed to be in column form, whereas the *transpose* of a vector, denoted

$\mathbf{v}^T$, represents its row form. Hence, the *inner product* of two vectors $\mathbf{v}$, $\mathbf{u}$ from the same real vector space is denoted $\mathbf{v}^T\mathbf{u}$. Matrix M is denoted by uppercase. In probabilistic computations often expressions will be encountered of the form $\mathbf{u} = \mathbf{v}M$ with $u_j = \sum_{i=1}^{n} v_i M_{i,j}$, $j = 1, \ldots, n$, and in that case both $\mathbf{u}$ and $\mathbf{v}$ are in row form, where multiplication is done according to the rules of matrix multiplication. It will become clear from the context whether use is made of notions from linear algebra (cf. e.g. [91]), or probability theory, as discussed in the following.

2.2 PROBABILITY THEORY

Now we continue with a brief review of some key concepts from probability theory, i.e., we consider events, joint probability distributions, conditional probability distributions, the chain rule, marginalization, and conditional independence. More detail about probability theory from a mathematical perspective is given in [33, 45]; an engineering approach is provided by [100].

Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ be a set of random variables, where $\text{Val}(X)$ indicates the *domain* of $X \in \mathbf{X}$ and $\text{Val}(\mathbf{X})$ the domain of $\mathbf{X}$, respectively. An (elementary) *event* $E \equiv X = x$ is any random variable X with a value x from its domain. The set of all possible Boolean combinations of events, or *Boolean algebra* denoted as $\mathcal{B}(\mathbf{X})$, is defined by using the operators: conjunction $(X = x \cap X' = x')$ (also called intersection), disjunction $(X = x \cup X' = x')$ (also called union), and negation $(\overline{X = x})$ (also called complementation). This Boolean algebra contains events such as $(X_1 = x_1 \cup X_2 = x_2)$, $(X_3 = x_3 \cap X_4 = x_4)$, and $\overline{X_2 = x_2}$. Events are partially ordered by $\subseteq$, with the universal lowerbound $\varnothing \in \mathcal{B}(\mathbf{X})$ and universal upperbound $\Omega \in \mathcal{B}(\mathbf{X})$, i.e., we have for each $E \in \mathcal{B}(\mathbf{X})$ that $\varnothing \subseteq E$ and $E \subseteq \Omega$. Usually $(X = x \cap X' = x')$ is represented in set notation as $\{X = x, X' = x'\}$.

A probability distribution is a function or mapping that assigns probabilities, i.e., values from the closed real interval $[0, 1]$, to any event involving variables in $\mathbf{X}$.

Definition 2.1 (Probability Distribution)**.** *A probability distribution for a set of random variables* $\mathbf{X}$ *with* domain $Val(\mathbf{X})$ *is defined as a function* $P : \mathcal{B}(\mathbf{X}) \to [0, 1]$*, such that the following axioms hold:*

(1) $P(E)$ *is a non-negative real value for all* $E \in \mathcal{B}(\mathbf{X})$*;*

(2) $P(\Omega) = 1$*;*

(3) *for any set of disjoint events* $E_1, \ldots, E_n \in \mathcal{B}(\mathbf{X})$, *with* $(E_i \cap E_j) = \varnothing$, $1 \leq i, j \leq n$, $i \neq j$, *we have that:*

$$P\left(\bigcup_{k=1}^{n} E_k\right) = \sum_{k=1}^{n} P(E_k).$$

It is a fundamental property of probability theory that it is sufficient to specify a probability distribution in terms of joint events $\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\}$, i.e., in terms of a *joint probability distribution* $P(X_1, X_2, \ldots, X_n)$ for all values of the domain $\text{Val}(\mathbf{X})$ (possibly with the exception of one element from $\text{Val}(\mathbf{X})$, where its probability can be derived from the other probabilities of elements of $\text{Val}(\mathbf{X})$ according to axioms (2) and (3)).

When the actual value of a random variable in an elementary event does not matter in a given context, we often also write $P(X)$ rather than $P(X = x)$ for the probability of variable X taking the value x.

The *marginal probability distribution* for a set of variables $\mathbf{Y}$ given the probability distribution for the random variables $\mathbf{X}$, with $\mathbf{Y} \subseteq \mathbf{X}$ and $\mathbf{X} = \mathbf{Y} \cup \mathbf{Z}$, where $\mathbf{Y}$ and $\mathbf{Z}$ are disjoint, is obtained by summing out the other variables (i.e. $\mathbf{Z}$) from the joint probability distribution $P(\mathbf{X})$, and is defined as:

$$P(\mathbf{Y}) = \sum_{\mathbf{z} \in \text{Val}(\mathbf{X} \setminus \mathbf{Y})} P(\mathbf{Y}, \mathbf{Z} = \mathbf{z})$$

Definition 2.2 (Conditional Probability Distribution)**.** *Let* $P(\mathbf{X}, \mathbf{Y})$ *be a joint probability distribution over a set of random variables* $\mathbf{X}$ *and* $\mathbf{Y}$*. A conditional probability distribution* $P(\mathbf{X} \mid \mathbf{Y})$ *is defined as:*

$$P(\mathbf{X} \mid \mathbf{Y}) = \frac{P(\mathbf{X}, \mathbf{Y})}{P(\mathbf{Y})} \tag{2.1}$$

with $P(\mathbf{Y}) > 0$*.*

It is good to realize that $P(\mathbf{X} \mid \mathbf{Y})$ is actually a family of probability distributions, one for every value $\mathbf{y}$ of $\mathbf{Y}$. The conditional probability $P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y})$ is the probability of the event $\mathbf{X} = \mathbf{x}$ given knowledge about the event $\mathbf{Y} = \mathbf{y}$.

The concept of conditional probability is one of the most fundamental and most important concepts in probability theory. In addition, the conditional probability plays an essential role in a wide range of domains, including classification, decision making, prediction and other

similar situations, where the results of interest are based on available knowledge.

By moving the denominator on the right of Equation 2.1 to the left, Equation 2.1 can also be written as:

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X} \mid \mathbf{Y})P(\mathbf{Y}) = P(\mathbf{Y} \mid \mathbf{X})P(\mathbf{X}) \tag{2.2}$$

By applying Equation 2.2 to a set of random variables $\{X_1, X_2, \ldots, X_n\}$, this creates a chain of conditional probabilities, more formally:

Proposition 2.2.1 (Chain Rule)**.** *Let P be a joint probability distribution over a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. Then it holds that:*

$$P(X_1, X_2, \ldots, X_n) = P(X_n \mid X_{n-1}, \ldots, X_1) \cdots P(X_2 \mid X_1)P(X_1)$$

The chain rule allows us to compute the joint distribution of a set of any random variables by only making use of conditional probabilities. This rule is particularly useful in Bayesian networks, which we will introduce later in this chapter. Combined with the network structures, the use of the chain rule can facilitate the representation for a joint distribution.

Another immediate result of Equation 2.2 by rearranging terms is *Bayes' rule*:

$$P(\mathbf{X} \mid \mathbf{Y}) = \frac{P(\mathbf{X})P(\mathbf{Y} \mid \mathbf{X})}{P(\mathbf{Y})}$$

Bayes' rule tells us how we can calculate a conditional probability given its inverse conditional probability. For example, using Bayes' rule makes it possible for us to derive the conditional probability $P(\mathbf{X} \mid \mathbf{Y})$ from its inverse conditional probability $P(\mathbf{Y} \mid \mathbf{X})$, if we also have information about the prior probability of events $\mathbf{X}$ and $\mathbf{Y}$.

A more general conditional version of Bayes' rule, where all probabilities are conditional on the same set of variables $\mathbf{Z}$, also holds:

$$P(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) = \frac{P(\mathbf{X} \mid \mathbf{Z})P(\mathbf{Y} \mid \mathbf{X}, \mathbf{Z})}{P(\mathbf{Y} \mid \mathbf{Z})}$$

with $P(\mathbf{Y} \mid \mathbf{Z}) > 0$.

Another fundamental concept in probability theory is *conditional independence*. Two sets of variables $\mathbf{X}$, $\mathbf{Y}$ are said to be conditionally independent given a set of variable $\mathbf{Z}$, denoted $\mathbf{X} \perp\!\!\!\perp_P \mathbf{Y} \mid \mathbf{Z}$, if

$$P(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z}) \quad \text{or} \quad P(\mathbf{Y}, \mathbf{Z}) = 0 \tag{2.3}$$

Equation 2.3 asserts that given knowledge of a set of variables $\mathbf{Z}$, knowledge of whether $\mathbf{Y}$ occurs provides no extra information on the probability of whether $\mathbf{X}$ occurs.

2.3 BAYESIAN NETWORKS

Bayesian networks are a compact and natural graphical representation of probability distributions. A Bayesian network, abbreviated as BN, is a probabilistic graphical model that represents a set of random variables and their conditional independences via a directed acyclic graph. For more details about Bayesian networks, we refer the reader to three books: the seminal work by Pearl [74], the encyclopedic account by Koller and Friedman [49], and the book by Neapolitan [66] that offers a good introduction to the underlying mathematics.

As Bayesian networks are a graphical formalism, we will use a lot of notions of graph theory and a small fraction of it is summarized next. Let the pair $G = (\mathbf{V}(G), \mathbf{E}(G))$ be a graph, often abbreviated to $G = (\mathbf{V}, \mathbf{E})$, then the set $\mathbf{V}$ is called its set of *nodes*, and the elements in the set $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ are called *edges*. For Bayesian networks, we restrict ourselves to *directed* edges or *arcs*, i.e., if $(v, v') \in \mathbf{E}$, then we assume that it is different from (v', v) and $(v', v) \notin \mathbf{E}$. An arc $(v, v') \in \mathbf{E}$ is often denoted $v \to v'$. When a graph G only contains arcs, it is called a *directed graph*. Furthermore, the concept of *children* of a node $v \in \mathbf{V}$ is defined as $\gamma(v) = \{v' \mid v \to v' \in \mathbf{E}\}$ and the set of *parents* of a node $v \in \mathbf{V}$ is defined as $\pi(v) = \{v' \mid v' \to v \in \mathbf{E}\}$. Finally, when we follow the arcs of a graph G between two nodes v and u we have a *directed path*; when there are *no* paths in the graph G of the form $v \to w \to \cdots \to u \to v$ (first and last node of the directed path are equal) it is called *acyclic*.

A formal definition for Bayesian networks is given in the following.

Definition 2.3 (Bayesian Network)**.** *A Bayesian network $\mathcal{B}$ is defined as a pair $\mathcal{B} = (G, P)$, where G is an acyclic directed graph and P a probability distribution. The graph $G = (\mathbf{V}, \mathbf{E})$, consists of a set of* nodes $\mathbf{V}$*, representing random variables, and a set of directed edges or arcs $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$. Let $X \in \mathbf{V}$ be a variable and $\pi(X)$ be the parents of X in graph G. The distribution P is defined as a joint distribution over variables $\mathbf{V}$, specified by multiplying*

conditional probability distributions for each variable $X \in \mathbf{V}$ *in the form of* $P(X \mid \pi(X))$*, formally:*

$$P(\mathbf{V}) = \prod_{X \in \mathbf{V}} P(X \mid \pi(X))$$

Example 2.1

Consider the problem of modeling *exacerbation* for a patient with chronic obstructive pulmonary disease (COPD), i.e., acute worsening events of COPD-related health status, using Bayesian networks. For this problem, we use the Bayesian network constructed based on expert knowledge and previously discussed in detail in [42]. The network is depicted in Fig. 2.1. The network contains two hidden variables, namely infection (I) and lung function (LF) which cannot be observed directly, but whose values can be derived based on indirect measures, such as body temperature for infection and the forced expiratory volume in 1 second (F) for lung function. Other important variables are the symptoms that one might expect a patient to report, such as dyspnea (D), sputum volume (V) and purulence (SP), cough (C), wheeze (W), malaise (M), temperature (T), whether performing daily activities is difficult due to COPD (A), and blood oxygen saturation (O).

The joint probability distribution P over the variables in the network is given by multiplying the conditional probabilities for each variable in the network, that gives:

$$\begin{aligned} &P(T, SP, SV, C, I, M, A, LF, D, E, W, O, F) \\ &\quad = P(T \mid I)P(SP \mid I)P(V \mid I)P(C \mid V)P(M \mid I)P(A \mid M, LF)P(E \mid LF) \\ &\quad\ P(F \mid LF)P(O \mid LF)P(W \mid LF)P(D \mid LF)P(I)P(LF \mid I) \end{aligned}$$

The standard interpretation of the graph of a Bayesian network is in terms of *d-separation*. When two of its disjoint sets of nodes $\mathbf{U}, \mathbf{W} \subseteq \mathbf{V}$ are connected by a path (ignoring the direction of the edges) that contains nodes v from a third, disjoint set $\mathbf{Z} \subseteq \mathbf{V}$ that only are serial nodes ($\cdots \to v \to \cdots$ or $\cdots \leftarrow v \leftarrow \cdots$) or divergent nodes ($\cdots \leftarrow v \to \cdots$), and none of the nodes v or its descendants in $\mathbf{Z}$ have two incoming arcs $\cdots \to v \leftarrow \cdots$, the path is called *blocked* given $\mathbf{Z}$. If every path between node sets $\mathbf{U}$ and $\mathbf{W}$ is blocked by $\mathbf{Z}$, it is said that $\mathbf{U}$ and $\mathbf{W}$ are *d-separated* given $\mathbf{Z}$ [74], often denoted as

$$\mathbf{U} \perp\!\!\!\perp_G \mathbf{W} \mid \mathbf{Z}.$$

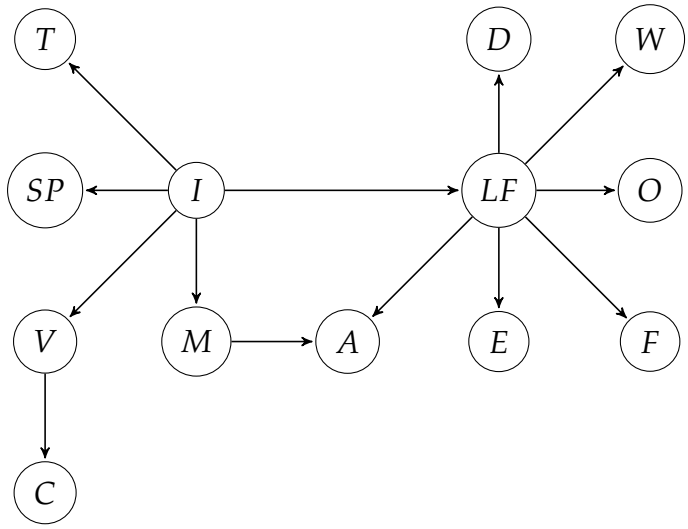


Fig. 2.1: Expert opinion based Bayesian network for COPD problem. A = activity, C = cough, D = dyspnea, E = exacerbation, F = FEV_1, I = infection, LF = lung function, M = malaise, O = oxygen saturation, SP = sputum purulence, V = sputum volume, T = temperature and W = Wheeze.

D-separation implies that the two corresponding variable sets **U** and **W** are conditionally independent given the variables corresponding to **Z**, i.e.,

$$\mathbf{U} \perp\!\!\!\perp_P \mathbf{W} \mid \mathbf{Z}$$

(note that we use the $\perp\!\!\!\perp_P$ relation here). The graph G of the Bayesian network $\mathcal{B} = (G, P)$ is said to be an independence map or *I-map* of P. If sets of nodes **U** and **W** are *not* d-separated given **Z**, they are called *d-connected*.

Example 2.2

Consider the network introduced in Example 2.1. According to standard interpretation of Bayesian networks, the graph tells us that D and W are d-connected given the empty set $\emptyset$, whereas they become d-separated given LF, as LF has a divergent connection on the path between D and W. In addition, M and LF are d-separated only given I but they become d-connected also given A, as A has two incoming arcs from M and LF.

2.4 MARKOV PROCESSES

Now we move on to modeling dynamic systems, where we are interested in reasoning about the state of a dynamic system as it evolves over time. The focus in this section is on modeling dynamic systems using *Markov processes* or *Markov chains*.

A Markov process is a stochastic process for which the future only depends on the present state. In other words, it has no memory of how the present state is reached. For Markov processes, we only consider a finite state space. Here we give a brief introduction of two widely used versions of Markov processes, discrete time Markov processes where time may be mapped to the natural numbers, and continuous time Markov processes where time may be mapped to the real numbers. Readers are encouraged to refer to more comprehensive materials (e.g. [36, 51]).

We use the notation $X_{[0,s)} \equiv \{X_t \mid 0 \leq t < s, \text{with } t, s \in \mathbb{R}_0^+\}$.

Definition 2.4 (Markov Property). *Let $\mathbf{X} = \{X_t \mid t \in \mathbb{R}_0^+\}$ be a stochastic process with X_t taking values from a finite set. The process $\{X_t \mid t \in \mathbb{R}_0^+\}$ is said to have the Markov property if the following condition is met:*

$$P(X_{s+t} \mid X_s, \mathbf{X}') = P(X_{s+t} \mid X_s), \quad s, t \in \mathbb{R}_0^+, \mathbf{X}' \subseteq X_{[0,s)}$$

with $\mathbf{X}'$ a countable set of instantiations.

Such a process $\{X_t \mid t \in \mathbb{R}_0^+\}$ is also called a *continuous time Markov process*. If the Markov property holds when the time is discrete, that is, time only takes values of natural numbers $\mathbb{N}_0$, the corresponding process $\{X_t \mid t \in \mathbb{N}_0\}$ is called a *discrete time Markov process*.

Markov chains are often assumed to be time-homogeneous, i.e., the probabilities of a variable X_t transitioning from its current state to its next state are the same, regardless of what the current time t is. Formally:

Definition 2.5 (Time Homogeneity). *Let $\{X_t \mid t \in \mathbb{R}_0^+\}$ be a continuous time Markov process. The chain $\{X_t \mid t \in \mathbb{R}_0^+\}$ is said to be a time homogeneous continuous time Markov chain if the following condition is met:*

$$P(X_{s+t} \mid X_s) = P(X_t \mid X_0), \quad \textit{for all } s, t \in \mathbb{R}_0^+ \tag{2.4}$$

If such a property also holds for a discrete time Markov process $\{X_t \mid t \in \mathbb{N}_0\}$, the process $\{X_t \mid t \in \mathbb{N}_0\}$ is called a time homogeneous discrete time Markov process.

In the case of discrete time, the transition probabilities $P(X_t \mid X_{t-1})$ assert the probabilities that variable X transitions from its state at time $t-1$ to another at time t. Let p_{ij} be the transition probability from state i at time $t-1$ to state j at time t. For a discrete time Markov process X with n possible states, transition probabilities for possible transitions between any two states are given via a *transition matrix* A, formally:

$$A_X = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

where $p_{ij} \geq 0$ and $\sum_j p_{ij} = 1$.

Example 2.3

Recall the variable V introduced in Example 2.1, which models whether a patient experiences an increase in sputum volume (v = increased, $\overline{v}$ = normal). A discrete time Markov chain can be employed to describe the dynamics of variable V at discrete time points by a transition matrix over its two states as given below:

$$A_V = \begin{array}{c} \begin{matrix} v & \overline{v} \end{matrix} \\ \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} \end{array} \begin{matrix} v \\ \overline{v} \end{matrix}$$

Let the time unit in this discrete time Markov chain be days; then the transition matrix states that a patient has a ninety percent chance to have increased sputum volume tomorrow given that there is an increase in sputum volume today, and there is a ninety percent chance that the sputum volume is normal tomorrow if its level is normal today.

In the case of continuous time, the dynamics of a continuous time Markov process X with n possible states are defined by an $n \times n$ intensity matrix:

$$Q_X = \begin{pmatrix} -q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & -q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & -q_{nn} \end{pmatrix} \tag{2.5}$$

where $q_{ij} \geq 0$ and $q_{ii} = \sum_{j \neq i} q_{ij}$. The reciprocal of the diagonal elements $1/q_{ii}$ gives the expected time that variable X will remain in the state i, and once it transitions, it shifts from state i to state j with probability q_{ij}/q_{ii}.

Example 2.4

Reconsider the problem of modeling the dynamics of variable V introduced in Example 2.1. We may want to know the state of variable V at any arbitrary time point, rather than only at discrete time points. Then the dynamics of variable V can be described by a continuous time Markov process, for example with the following intensity matrix:

$$Q_V = \begin{array}{c} \\ \overline{v} \\ v \end{array} \begin{array}{c} \begin{array}{cc} \overline{v} & v \end{array} \\ \begin{pmatrix} -2 & 2 \\ 1 & -1 \end{pmatrix} \end{array}$$

Here the entry $(2, 2)$ indicates that we expect on average a patient will experience an increase in the sputum volume in a half hour ($1/q_{22} = 1/2$), if the patient's sputum volume currently is normal and we view units for time as hours.

2.5 TEMPORAL BAYESIAN NETWORKS

In the previous section, we described how to model dynamic systems using Markov processes. Now we present two formulations based on

Markov processes to describe structured stochastic processes: dynamic Bayesian networks (DBNs) [14, 64] and continuous time Bayesian networks (CTBNs) [71]. These models represent a stochastic process over a *structured state space* consisting of assignments to a set of local variables. The dynamics of the temporal evolution of the structured state space are described in terms of the evolution of the local variables.

DBNs and CTBNs extend standard Bayesian networks by assuming that changes in a process can be captured by a sequence of states at either *discrete* or *continuous* time points. In both DBNs and CTBNs, time t is modeled explicitly. Dynamic Bayesian networks are based on discrete time Markov processes that describe structured stochastic processes at discrete time points, while CTBNs are based on continuous time Markov processes. We use X_t to represent the instantiation of variable X at time t. In CTBNs for simplicity's sake, we will also use the notation X_I with $I \subseteq \mathbb{R}_0^+$ a real interval (closed $I = [l, u]$, open $I = (l, u)$, or half-open $I = (l, u]$ or $I = [l, u)$), where in particular $X_{[l,u)}$ denotes the set of instantiations of variable X_t in the half-open time interval $t \in [l, u)$, i.e., $\{X_t \mid l \leq t < u, \text{with } l, t, u \in \mathbb{R}_0^+\}$. For DBNs, if the time index t comes from a sequence of time points $\{0, 1, \ldots, T\} = 0 : T$, $T \in \mathbb{N}_0$, we use the notation $X_{0:T}$.

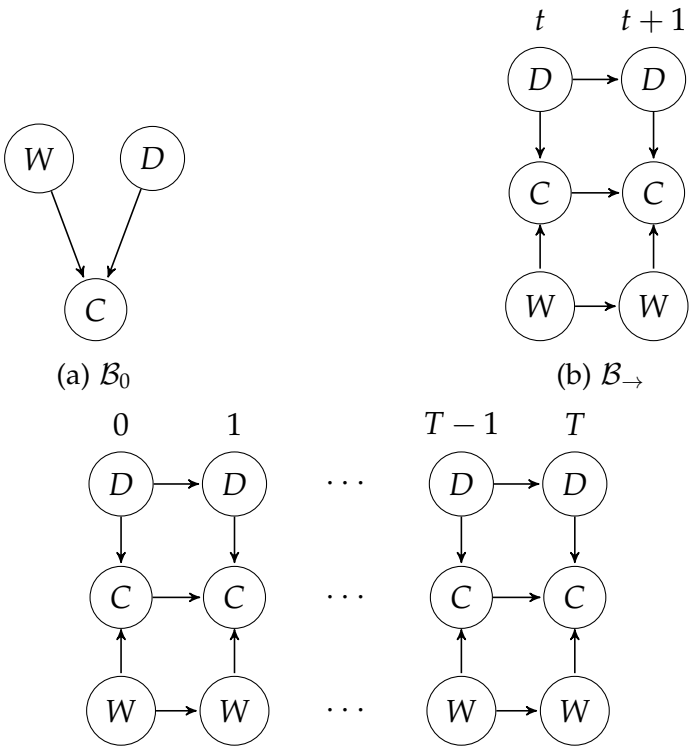
2.5.1 *Dynamic Bayesian Networks*

Definition 2.6 (Dynamic Bayesian Network)**.** *A dynamic Bayesian network is defined as a pair $(\mathcal{B}_0, \mathcal{B}_{\rightarrow})$ over variables $\mathbf{V}$, where $\mathcal{B}_0$ is a Bayesian network over variables $\mathbf{V}_0$ representing the* initial distribution *over states, and $\mathcal{B}_{\rightarrow}$ is defined as a* conditional distribution *for a* 2-time-slice Bayesian network *(2-TBN) given by:*

$$P(\mathbf{V}_{t+1} \mid \mathbf{V}_t) = \prod_{X \in \mathbf{V}} P(X_{t+1} \mid \pi(X_{t+1}))$$

for every time-point $t \in \mathbb{N}_0$, where X_t is the instantiation of variable X at time t, and the variables $\mathbf{V}_t$ are the instantiations of a set of variables $\mathbf{V}$ at time t.

The parent set $\pi(X_{t+1})$ may include variables from the time slice $t + 1$, with outgoing *intra-time-slice arcs*, or the previous time slice t, with outgoing *inter-time-slice-arcs* in the 2-TBN. The conditional probabilities $P(X_{t+1} \mid \pi(X_{t+1}))$ in the 2-TBN do not vary over time, i.e., are *stationary*. In addition, the Markov property holds for DBNs.



(a) $\mathcal{B}_0$ (b) $\mathcal{B}_{\rightarrow}$

(c) DBN unrolled over T steps

Fig. 2.2: A simplified DBN with three variables from the COPD network given in Fig. 2.1 : (a) the initial model; (b) the 2-TBN; (3) the resulting unrolled DBN over T steps.

For any desired time span $T \geq 0$, the joint distribution over $\mathbf{V}_{0:T}$, the set of variables $\{\mathbf{V}_t \mid t \in 0:T\}$, is defined by a product of the conditional probability distributions in the initial model and in the 2-TBN:

$$P(\mathbf{V}_{0:T}) = \prod_{X \in \mathbf{V}} P_{\mathcal{B}_0}(X_0 \mid \pi(X_0)) \prod_{X \in \mathbf{V}} \prod_{t \in 0:T-1} P_{\mathcal{B}_{\rightarrow}}(X_{t+1} \mid \pi(X_{t+1})) \qquad (2.6)$$

We can obtain a standard Bayesian network by 'unrolling' the 2-TBN over T steps.

Example 2.5

Consider a dynamic Bayesian network over variables $\mathbf{V}$, $\mathbf{V} = \{D, W, C\}$, with an initial model in Fig. 2.2a and a 2-TBN in Fig. 2.2b. For the variable C_{t+1} in the 2-TBN, we have intra-time-slice arcs $W_{t+1} \rightarrow C_{t+1}$ and

$D_{t+1} \rightarrow C_{t+1}$, and inter-time-slice arc $C_t \rightarrow C_{t+1}$. The DBN can be unrolled over T steps, resulting in a standard Bayesian network as shown in Fig. 2.2c. The joint distribution for the standard Bayesian network is given:

$$P(\mathbf{V}_{0:T}) = P(D_0)P(W_0)P(C_0 \mid D_0, W_0) \prod_{t=0}^{T-1} P(D_{t+1} \mid D_t)P(W_{t+1} \mid W_t)P(C_{t+1} \mid D_{t+1}, W_{t+1}, C_t)$$

2.5.2 *Continuous-time Bayesian Networks*

Continuous-time Bayesian networks (CTBNs) are specified in terms of variables with temporal dynamics given by an *intensity matrix* as defined in Equation 2.5. In CTBNs, a central concept for a variable X is its *conditional intensity matrix* (CIM), denoted by $Q_{X|\pi(X)}$, specifying an intensity matrix for variable X for each value of its parents $\pi(X)$. The intensities governing the dynamics of process X vary over time. However, the intensities are not represented as a function of time, but as a function of the current values of its parents $\pi(X)$.

Definition 2.7 (Continuous-time Bayesian Network)**.** *A continuous-time Bayesian network (CTBN) is defined as a tuple $\mathcal{N} = (\mathcal{B}, G_{\rightarrow}, \mathbf{Q})$, where $\mathcal{B} = (G_0, P)$ denotes the Bayesian network that specifies the initial model; $G_{\rightarrow}$ denotes the graph of the transition model and $\mathbf{Q}$ is a set of intensity matrices for all the variables in the CTBN. The graph of a CTBN is $(G_0, G_{\rightarrow})$, with $G_0 = (\mathbf{V}(G_0), \mathbf{E}(G_0))$ and $G_{\rightarrow} = (\mathbf{V}(G_{\rightarrow}), \mathbf{E}(G_{\rightarrow}))$.*

Example 2.6

Consider a variable C which models whether a patient coughs or not (c: cough, $\overline{c}$: not cough), and is conditioned on a variable V which models whether a patient has an increased sputum volume. Then we can specify two CIMs for variable C corresponding to each value of variable V as below:

$$Q_{C|v} = \begin{array}{c} \\ \overline{c} \\ c \end{array} \begin{array}{c} \begin{array}{cc} \overline{c} & c \end{array} \\ \begin{pmatrix} -4 & 4 \\ 3 & -3 \end{pmatrix} \end{array} \qquad Q_{C|\overline{v}} = \begin{array}{c} \\ \overline{c} \\ c \end{array} \begin{array}{c} \begin{array}{cc} \overline{c} & c \end{array} \\ \begin{pmatrix} -6 & 6 \\ 5 & -5 \end{pmatrix} \end{array}$$

In order to compose Markov processes in a larger network, a 'multiplication' operation called *amalgamation* is defined in CTBNs, which takes two conditional intensity matrices and forms from them a joint intensity matrix. For the amalgamation, an assumption is made that the intensities in the resulting intensity matrix corresponding to two simultaneous changes are zeros.

Before we perform an amalgamation, we first need a mapping between row or column numbers in the joint intensity matrix and instantiations of the variables. We first define a binary total order relation, denoted by $\prec$. Given an order on a finite set of variables $\mathbf{V}$, $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$, and on the states of each variable $\mathrm{Val}(X_i), i \in \{1, \ldots, n\}$, we say $(x_1, \ldots, x_n) \prec (x'_1, \ldots, x'_n)$, if $x_n < x'_n$, or $(x_1, \ldots, x_{n-1}) \prec (x'_1, \ldots, x'_{n-1})$ and $x_n = x'_n$.

Example 2.7

Consider two variables V and C with each two ordered states, $\{v, \overline{v}\}$ for variable V and $\{c, \overline{c}\}$ for variable C. Given variables V and C, the variable order $V < C$ and the order on their states, $\overline{v} < v$ and $\overline{c} < c$, we can obtain a sequence of states over variables V, C, i.e., $\langle \overline{v}, \overline{c} \rangle, \langle v, \overline{c} \rangle, \langle \overline{v}, c \rangle, \langle v, c \rangle$.

For a set of N variables with each variable having n_i states, we have $n = \prod_{i=1}^{N} n_i$ possible joint states and the size of the resulting joint intensity matrix is $n \times n$. As the number of variables increases, the size of the joint intensity matrix grows exponentially. To compute the joint intensity matrix, conditional intensity matrices for each variable have to be expanded first to the size of the resulting joint intensity matrix. A detailed discussion of how to expand these intensity matrices is given in [68].

Given the variable ordering defined in Example 2.7, we now can perform the amalgamation operation over variable V and C.

Example 2.8

Consider we have a CTBN with structure $V \rightarrow C$ with conditional intensity matrices given for variable V in Example 2.4 and for variable C in Example 2.6. We obtain a 4×4 intensity matrix Q_{VC} from the 2×2

conditional intensity matrices Q_V for variable V and $Q_{C|V}$ for variable C:

$$Q_{VC} = \begin{array}{c} \begin{matrix} \langle \overline{v}, \overline{c} \rangle & \langle v, \overline{c} \rangle & \langle \overline{v}, c \rangle & \langle v, c \rangle \end{matrix} \\ \begin{pmatrix} -8 & 2 & 6 & 0 \\ 1 & -5 & 0 & 4 \\ 5 & 0 & -7 & 2 \\ 0 & 3 & 1 & -4 \end{pmatrix} \end{array} \begin{matrix} \langle \overline{v}, \overline{c} \rangle \\ \langle v, \overline{c} \rangle \\ \langle \overline{v}, c \rangle \\ \langle v, c \rangle \end{matrix}$$

For a homogeneous Markov process over variables $\mathbf{V}$ with an intensity matrix $Q_{\mathbf{V}}$ and an initial distribution $P(\mathbf{V}_0)$, we can compute the distribution over the values of variables $\mathbf{V}$ at a particular time point or the joint distribution at different time points. The joint distribution at a finite set of time points of interest $\mathbf{B}$ is given by:

$$\mathbf{P}(\mathbf{V}_{\mathbf{B}}) = \mathbf{P}(\mathbf{V}_0) \prod_{t \in \mathbf{B} \setminus \max(\mathbf{B})} \exp(Q_{\mathbf{V}}(s(t) - t)) \tag{2.7}$$

where $\mathbf{P}(\mathbf{V}_0)$ and $\mathbf{P}(\mathbf{V}_{\mathbf{B}})$ are probability row vectors, the symbol exp is the matrix exponential that is defined in terms of its Taylor series expansion:

$$\exp A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \tag{2.8}$$

Rarely is it the case that the matrix exponential can be computed easily; one well-known example where this is the case is a diagonal matrix, where one can simply take the exponential of the individual diagonal elements to obtain the matrix exponential.

2.6 MODEL LEARNING

As Bayesian networks are based on probability theory, statistical methods can be used to infer these models from data in the presence of noise and uncertainty. Data can be used to restore solely the underlying graph structure, or to estimate the parameters in a model given a predetermined graph structure, or to learn both structure and parameters. In this section, we give a brief overview of basic methods to learn structure and parameters for Bayesian networks.

2.6.1 *Structure Learning*

One approach to learning an unknown directed graph structure is called *constraint-based structure learning*. This approach views a Bayesian network as a representation of independences. The goal of structure learning is to find a network that best explains the independences and dependences by performing statistical conditional independence tests. A fundamental question here is to determine whether two variables are independent given a third set of variables, for which various *independence testing* methods are being used.

An alternative approach is score-based structure learning. This approach looks upon a Bayesian network as specifying a statistical model and then addresses learning as a *model selection* problem. It considers all possible candidate structures and selects one candidate structure that best fits data based on a scoring function. The number of candidate models grows more than exponentially with the number of variables.

2.6.2 *Parameter Learning*

The essence of parameter learning is to find the maximum likelihood estimate, or maximum a posteriori estimation when a prior distribution is also considered during parameter learning. Learning parameters from complete data, i.e., when values for all variables in a model are known, and from incomplete data, i.e., values for some variables are missing, are different. In the former, sufficient statistics, i.e., the counts of instantiations of variables in the data, are sufficient to derive the desired parameters, whereas additional algorithms are needed to handle missing values in the latter. Here we review two common methods, i.e., the expectation maximization algorithm (EM) and the Markov chain Monte Carlo (MCMC) method, to handle missing values. The former is used in Chapter 4 and 5, and the latter is employed to estimate parameters from incomplete data in Chapter 4.

EXPECTATION MAXIMIZATION ALGORITHM The Expectation Maximization Algorithm, or EM algorithm for short, is a general framework of maximum likelihood estimation in the presence of hidden variables or missing values in data. Essentially, it is an iterative procedure to find the maximum likelihood estimate in statistical models. The EM algorithm alternates between performing an Expectation (E) step, which cre-

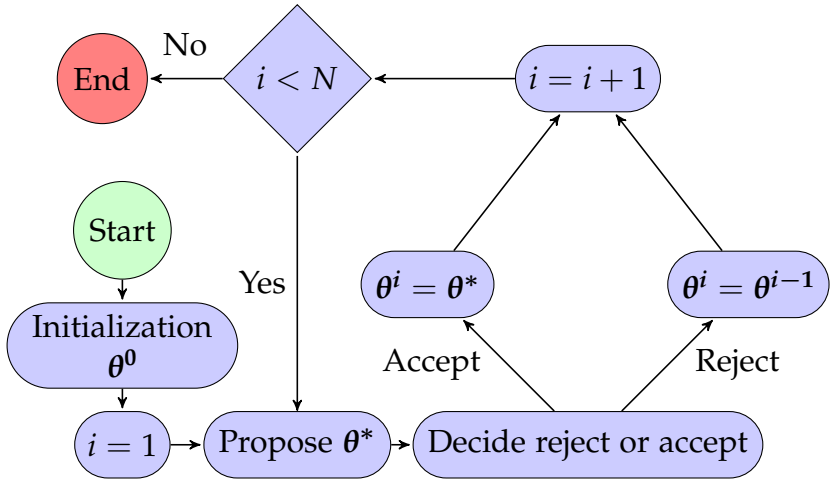


Fig. 2.3: Flow chart of an MCMC procedure with θ standing for the desired parameters and N the required number of iterations.

ates an expected log-likelihood function using the current estimate for the parameters, and a maximization (M) step, which derives new parameters by maximizing the expected log-likelihood obtained on the E step. The EM algorithm repeats E and M steps until the parameters converge after a certain number of iterations. However, it is not guaranteed that the global maximum will be reached, as a different starting value may end up at a local maximum.

MARKOV CHAIN MONTE CARLO METHOD The Markov Chain Monte Carlo method, or MCMC for short, is a generic framework for generating samples from a posterior distribution, in cases where we cannot efficiently sample from the posterior distribution directly. For example, the posterior distribution is high-dimensional or cannot be written in a closed-form for mathematical analysis. In MCMC, a Markov chain is constructed which has the desired posterior distribution as its stationary distribution by iteratively generating samples that are closer and closer to the posterior. This method has played a significant role in estimating parameters for Bayesian model from data containing missing values, which is a common problem in machine learning.

A flow chart of a general MCMC framework is given in Fig. 2.3. To summarize, the initial values of parameters θ are chosen randomly to start a Markov chain, and set iteration $i = 1$. For each iteration i, a candidate θ^* is sampled from a distribution, which can be any proposal dis-

tribution, as often seen in the Metropolis-Hastings algorithm, a widely used implementation of the MCMC framework. The candidate $\boldsymbol{\theta}^*$ is decided to be rejected or not by a selection criterion. In the Metropolis-Hastings algorithm, for example, it is determined by the probability acceptance, which indicates how probable the candidate sample is with respect to the current sample, according to the desired probability distribution. The process continues generating samples in this way until the number of iterations is exceeded or sufficient number of samples are collected (Note that the latter is omitted in the flow chart in Fig. 2.3 for simplicity).

There are also some challenges in the applicability of the MCMC methods. In practise, the MCMC methods are not generally an out-of-box solution, and many options need to be specified: the proposal distribution, the number of iterations to run, the metrics for evaluating mixing, etc. In addition, the stationary distribution of the constructed Markov chain is regarded as the desired posterior distribution. However, this is only guaranteed in the limit in theory. In practice, diagnostics must be applied to monitor whether the Markov chain has converged.

3

HYBRID TIME BAYESIAN NETWORKS

Capturing the behavior of a heterogeneous dynamic system in a probabilistic model is a challenging problem. A single time granularity, such as employed by dynamic Bayesian networks, provides insufficient flexibility to capture the dynamics of many real-world processes. An alternative is to assume that time is continuous, giving rise to continuous time Bayesian networks. Here the problem is that the level of temporal detail is too precise to match available probabilistic knowledge. In this chapter, a novel class of probabilistic models is presented, called hybrid time Bayesian networks; they combine discrete-time and continuous-time Bayesian networks. The new formalism allows us to more naturally model dynamic systems with regular and irregularly changing variables. We also present a mechanism to construct discrete-time versions of hybrid models and an expectation-maximization-based (EM-based) algorithm to learn the parameters of the constructed models. The usefulness of the proposed models is illustrated by means of a real-world medical problem.

3.1 INTRODUCTION

Many real-world systems exhibit complex and rich dynamic behavior. As a consequence, capturing these dynamics is an integral part of developing models of physical-world systems. Representing dynamics always involves the parameter of *time*. Often variation in behavior is captured by variables that are indexed by time. In order to represent the overall behavior of systems, it is often assumed that the time indices are equally spaced, for example as natural numbers, or take all elements of the non-negative real axis. However, if we are dealing with real-world observations, time indices are usually not equally spaced. This issue is described by *time granularity*, i.e., the actual difference in time between subsequent events.

Time granularity is an important parameter in characterizing dynamics as it determines the level of temporal detail in the model. In cases where one time granularity is coarser than another, dealing with multiple time granularities becomes significantly important, e.g., in the context of mining frequent patterns and temporal relationships in data streams and databases [7].

Bayesian networks (BNs) have been very successful in modeling complex situations involving uncertainty [74]. Dynamic Bayesian networks (DBNs) are part of the Bayesian network framework, supporting the modeling of dynamic probabilistic systems [64]. DBNs extend standard Bayesian networks by assuming that changes in a process can be captured by a sequence of states at discrete time points. Usually the assumption is 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 temporal part of the process changes much faster than another. In that case, the whole system has to be represented using the finest time granularity, which is undesirable from a modeling and learning perspective. In particular, if a variable is observed irregularly, much data on discrete-time points will be missing and conditional probabilities will be hard to estimate.

As an alternative to DBNs, temporal processes can be modeled as continuous time Bayesian networks (CTBNs), where time acts as a continuous parameter [71]. In these models, the time granularity is infinitely small by modeling transition rates rather than conditional probabilities. Thus, multiple time granularities, i.e., slow and fast transition rates, can easily be captured. A limitation from a modeling perspective is that all probabilistic knowledge, for example derived from expert knowledge, has to be mapped to transition rates which are hard to interpret. Moreover, it is assumed that the transition time, i.e., the time until a transition occurs, is exponentially distributed, which may not always be appropriate.

In this chapter, we propose a new formalism, which we call *hybrid time Bayesian networks* (HTBNs), inspired by discrete-time and continuous-time Bayesian networks. We develop the theoretical properties of HTBNs and show their practical use by means of a medical example. HTBNs facilitate modeling the dynamics of both irregularly-timed random variables and random variables whose evolution is naturally

described by discrete time. As a result, the new formalism increases the modeling and analysis capabilities for dynamic systems.

In the next section we introduce the clinical running example for the rest of this chapter. Then, in Section 3.3, we define HTBNs with their associated factorization, followed by a construction that allows transforming an HTBN into an equivalent BN. Subsequently, we return to our running example and demonstrate how the equivalent BN can be used to obtain a meaningful clinical simulation. The chapter is concluded by a discussion.

3.2 MOTIVATING EXAMPLE

To illustrate the usefulness of the proposed theory, we consider the medical problem of heart failure and, in particular, one possible cause of heart failure: heart attack (myocardial infarction). This usually occurs as the result of coronary artery disease giving rise to reduced blood supply to the heart muscle (myocardium). 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 heart muscle proteins, in particular troponin. Loss of heart muscle will inevitably have an impact on the contractability of the myocardium, and thus heart function will be negatively affected. This is known as *heart failure*. In particular, the heart fails with respect to its function as a pump. This will enforce an increase in the amount of extracellular fluid (the patient is flooded with water), which can be measured quite simply by means of the body weight. With regard to treatment, digitalis is considered as one of the drugs to improve contractability. This causal knowledge is formalized as a directed graph in Fig. 3.1.

Heart attacks can occur repeatedly in patients, although after some interval of time, and this may negatively affect heart function. After administration of digitalis it will take some time before the drug has a diminishing effect on heart failure. Thus, the course of heart failure will likely depend on various factors, and how they interact. Of particular importance here is the dynamics of the probability distributions over time.

In modeling processes such as heart failure, it is essential to notice the existence of different time granularities. There are *discrete*, *regular* variables which are observed regularly such as a routine checkup for body weight and a regular intake of a drug. On the other hand, some

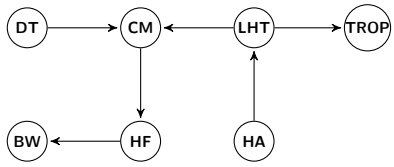


Fig. 3.1: Causal model for heart failure: CM = contractility myocardium, DT = digitalis, LHT = loss heart tissues, HA = heart attack, TROP = troponin, HF = heart failure, BW = body weight.

variables are observed *irregularly*, such as the indicator troponin which is elevated after about half an hour after damage to the heart muscle is obtained; however its measurement is repeated with time intervals that increase after the patient's condition has been stabilized. Clearly, it is not possible to obtain a satisfactory representation of the clinical evolution of heart failure using only discrete time, regular or irregular, or continuous time. In the remainder of this chapter we propose a method to deal with these heterogeneous time aspects.

3.3 HYBRID TIME BAYESIAN NETWORKS

In this section, we define hybrid-time Bayesian networks and give the semantics of these models in terms of their factorization. In the reminder of this chapter, symbol $\mathbf{A}$ denotes a set of time points, with $\mathbf{A} = \{0, 1, \ldots, n\}$ and symbol $\mathbf{B}$ a set of time points with $\mathbf{A} \subseteq \mathbf{B} \subset \mathbf{R}_0^+$.

3.3.1 *General Idea*

Random variables model dynamical systems evolving in discrete time steps or smoothly over time. DBNs are a general framework for modeling dynamic probabilistic systems on a set of discrete time points. The dynamics are parameterized by a conditional probability distribution of variables at time $s(\alpha)$, with s the successor function as defined in Section 2.1, conditioned on variables at time α. Building a DBN requires finding a single time granularity, though variables change at their own rate. When the modeled time granularity is coarser than the time spent between two consecutive observations, the conditional probabilities for some observations between discrete time slices are difficult to estimate, that is, it can give rise to information loss. In contrast, when time is continuous, intensity matrices in CTBNs are used to describe transient

behavior of random variables. Transition probabilities can be induced from intensity matrices, yet, these probabilities cannot be represented explictly in CTBNs. This implies that probabilistic knowledge from domain experts has to be mapped to intensity matrices which are hard to interpret. As a result, the question arises whether it is possible to have a probabilistic representation modeling dynamical systems with multiple changing rates. One answer is to define a probabilistic representation for modeling both discrete-time and continuous-time variables.

Dependences of random variables are represented by arcs in probabilistic graphical models, such as DBNs and CTBNs. In discrete-time models one can distinguish arcs which are inter-time-slice (also known as temporal arcs), going between time slices, or intra-time-slice (called atemporal arcs), connecting variables within the same time slice. The decision on how to relate two variables is dependent on how tight the coupling between them is in time. If the dependence of one variable with another is immediate, that is, much shorter than the time granularity in the model, an atemporal arc is appropriate. If the dependence manifests after some time but longer than the modeled time granularity, this can be modeled as an arc from one slice to the next. In contrast, arcs in CTBNs are always temporal: the dependence between variables manifests itself in the probability distribution of states mediated by an intensity matrix. The time it takes for a variable to influence another can be arbitrarily small, yet non-zero, which we denote by ϵ.

In these two modeling approaches, the common factor is that the dependences manifest a *delay*. In DBNs dependences have delay zero represented as atemporal arcs or some non-zero delay for temporal arcs, whereas in CTBNs all arcs have the same delay ϵ. For our definition of hybrid-time Bayesian networks, we can therefore incorporate continuous-time and discrete-time BNs in a single graph by assigning a delay to each arc, capturing the dependence behavior over time in both DBNs and CTBNs.

3.3.2 *Model Definition*

To define hybrid-time models, we first formalize the concept of arcs that represent delayed dependence. To this end, we define an arc with an associated attribute *delay*, explicitly indicating when the influence between variables manifests. The value of a delay is either a natural number d, $d \in \mathbb{N}_0$, or a small real number ϵ, $\epsilon \in \mathbb{R}_0^+$, $\epsilon \downarrow 0$. Atemporal

and temporal dependences in DBNs are represented by arcs having a natural number as delay of either $d = 0$ (atemporal) or $d \in \{1, 2, \ldots\}$ (temporal), respectively. The dependences in CTBNs are represented by arcs with a delay having the value ϵ.

Now, we will give a formal definition of arcs in a hybrid-time graph. Given a graph $G = (\mathbf{V}(G), \mathbf{E}(G))$, the arcs are defined as $\mathbf{E}(G) \subseteq \mathbf{V}(G) \times \mathbf{V}(G) \times (\mathbb{N}_0 \cup \{\epsilon\})$, each of which represents direct dependence of one variable on another.

The formal definition of hybrid time Bayesian networks is given as follows.

Definition 3.1 (Hybrid Time Bayesian Networks (HTBNs))**.** *A hybrid time Bayesian network is a tuple $\mathcal{H} = (\mathcal{B}, G_{\mathcal{H}}, \Phi, \mathbf{Q})$, where $\mathcal{B} = (G_0, P)$ is a Bayesian network specifying an initial distribution, $G_{\mathcal{H}} = (\mathbf{V}(G_{\mathcal{H}}), \mathbf{E}(G_{\mathcal{H}}))$ is a directed graph specifying a transition model with each node in $\mathbf{V}(G_{\mathcal{H}})$ 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 $\mathbf{Q}$ is a set of conditional intensity matrices for variables $\mathbf{C}$. Furthermore, graph $G_{\mathcal{H}}$ has the following properties:*

1. *An arc to a continuous-time variable has a delay ϵ;*
2. *An arc between discrete-time variables has a delay d, $d \in \mathbb{N}_0$;*
3. *An arc from a continuous-time variable to a discrete-time variable has delay 0;*
4. *$(\mathbf{V}(G_{\mathcal{H}}), \{(x, y) \mid (x, y, 0) \in \mathbf{E}(G_{\mathcal{H}})\})$ is acyclic.*

We use natural numbers $d \in \mathbb{N}_0$ to achieve generality of the representation of delays. However, as we restrict ourselves to first-order Markov models in this chapter, we will only deal with the subset $\{0, 1\}$ of the natural numbers $\mathbb{N}_0$ for delays in DBNs.

A cycle of arcs with delay 0 is not permitted, analogous to the requirement for BN graphs to be acyclic. A cycle containing arcs with non-zero delay, however, is allowed as an inherited property from discrete-time and continuous-time Bayesian networks. Although in principle arbitrary delays are possible, we here restrict ourselves to temporal dependences between continuous time variables with delay ϵ.

Example 3.1

In the example discussed in Section 3.2, regular variables, i.e., BW, DT,

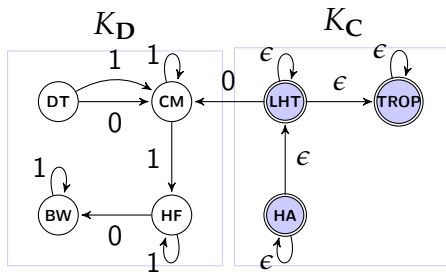


Fig. 3.2: A transition model of an HTBN for the heart failure problem. Continuous-time variables are graphically represented by double-edged shaded circles; the other circles indicate discrete-time variables.

HF and the hidden variable CM can be represented in a discrete-time manner. The irregular variables, i.e., LHT, TROP, HA are modeled as continuous-time variables. The example is then represented in a hybrid time Bayesian network $\mathcal{H}$ as shown in Fig. 3.2.

3.3.3 *Factorization*

The joint probability distribution for hybrid time Bayesian networks is defined by the multiplication of the conditional joint probabilities for the continuous-time and discrete-time Bayesian network parts. To this end, we first need to introduce some theoretical graph notions.

The *skeleton* $G^{\sim}$ of a directed graph G is obtained by changing the arcs in G to (undirected) edges. Every directed graph can be defined as the union of *connected components* by an equivalence relation $X - Y$, meaning that node Y can be reached by an undirected path from node X in its skeleton. Nodes X and Y are then members of the same equivalence class $[X]$ and the corresponding graph is a connected component. A graph $G' = (\mathbf{V}(G'), \mathbf{E}(G'))$ is said to be a *continuous-time induced subgraph* of G, denoted as $G_{\mathbf{C}}$, if $\mathbf{V}(G') = \mathbf{C}$ and $\mathbf{E}(G') = (\mathbf{V}(G') \times \mathbf{V}(G') \times \{\epsilon\}) \cap (\mathbf{V}(G) \times \mathbf{V}(G) \times \{\epsilon\})$. Similarly, G' is a *discrete-time induced subgraph* of G, denoted as $G_{\mathbf{D}}$, if $\mathbf{V}(G') = \mathbf{D}$ and $\mathbf{E}(G') = (\mathbf{V}(G') \times \mathbf{V}(G') \times \{0,1\}) \cap (\mathbf{V}(G) \times \mathbf{V}(G) \times \{0,1\})$.

Both $G_{\mathbf{C}}$ and $G_{\mathbf{D}}$ can be decomposed into connected components; each individual connected component is indicated by $K_{\mathbf{C}}$ and $K_{\mathbf{D}}$, respectively. Clearly connected components are disjoint as they represent equivalence classes and together the connected components form partitions of the continuous-time and discrete-time subgraphs, respectively.

A subset $\mathbf{X} \subseteq \mathbf{V}(G_{\mathbf{D}})$ is said to constitute the parents of $\mathbf{V}(K_{\mathbf{C}})$, denoted as $\pi(\mathbf{V}(K_{\mathbf{C}}))$, if and only if there exists at least an arc (D, C) in G, $C \in \mathbf{V}(K_{\mathbf{C}})$, for every $D \in \mathbf{X}$. Parents $\pi(\mathbf{V}(K_{\mathbf{D}}))$ are defined analogously.

Example 3.2

In the graph shown in Fig. 3.2, there is only one continuous-time connected component with $V(K_{\mathbf{C}}) = \{\text{LHT}, \text{TROP}, \text{HA}\}$ and one discrete-time connected component with $V(K_{\mathbf{D}}) = \{\text{DT}, \text{CM}, \text{HF}, \text{BW}\}$.

We are now in the position to define a conditional distribution of connected components given their parents.

Definition 3.2 (Conditional Joint Distribution for Component $K_{\mathbf{D}}$)**.** *Given a discrete-time component $K_{\mathbf{D}}$, the conditional joint distribution for $K_{\mathbf{D}}$ over time points of interest $\mathbf{A}$ is defined as:*

$$P(\mathbf{V}(K_{\mathbf{D}})_{\mathbf{A}} \mid \pi(\mathbf{V}(K_{\mathbf{D}}))_{\mathbf{A}}) = \prod_{D \in \mathbf{V}(K_{\mathbf{D}})} (P(D_0 \mid \pi(D)_0) \prod_{\alpha \in \mathbf{A} \setminus \{0\}} P(D_\alpha \mid \pi(D; \alpha)))$$

where $\pi(D; \alpha) = \{\pi(D)_{\alpha - d} \mid d \in \{0, 1\}\}$.

Definition 3.3 (Conditional Joint Distribution for Component $K_{\mathbf{C}}$)**.** *Given a continuous-time component $K_{\mathbf{C}}$ over variables $\mathbf{V}(K_{\mathbf{C}})$ with an initial distribution $P(\mathbf{V}(K_{\mathbf{C}})_0)$ and corresponding parents $\pi(\mathbf{V}(K_{\mathbf{C}}))$ over time points of interest $\mathbf{A}$. The conditional joint distribution for $K_{\mathbf{C}}$ over a finite set of time points of interest $\mathbf{B}$, with $0 \in \mathbf{A} \subseteq \mathbf{B} \subset \mathbb{R}_0^+$, is defined as:*

$$P(\mathbf{V}(K_{\mathbf{C}})_{\mathbf{B}} \mid \pi(\mathbf{V}(K_{\mathbf{C}}))_{\mathbf{A}}) = P(\mathbf{V}(K_{\mathbf{C}})_0) \prod_{\beta \in \mathbf{B} \setminus \{\max \mathbf{B}\}} \exp(Q_{\mathbf{V}(K_{\mathbf{C}}) \mid \pi(\mathbf{V}(K_{\mathbf{C}}))_a}(s(\beta) - \beta)) \quad (3.1)$$

with $a = \max\{\alpha \mid \alpha < \beta, \alpha \in \mathbf{A}\}$, and $Q_{\mathbf{V}(K_{\mathbf{C}}) \mid \pi(\mathbf{V}(K_{\mathbf{C}}))_a}$ is the conditional intensity matrix for variables $\mathbf{V}(K_{\mathbf{C}})$ given the values of parents $\pi(\mathbf{V}(K_{\mathbf{C}}))$ at time a.

Now we can define the full joint probability distribution of a hybrid-time BN given sets of time points of interest.

Definition 3.4 (Joint Probability Distribution)**.** *Given a hybrid time Bayesian network $\mathcal{H}$ and sets of components $\boldsymbol{K}_{\mathbf{D}}$, $\boldsymbol{K}_{\mathbf{C}}$ with associated time points of interest $\mathbf{A}$, $\mathbf{B}$. The joint distribution for $\mathcal{H}$ over $\mathbf{B}$ is defined as:*

$$P(\mathbf{V}(G)_{\mathbf{B}}) = \prod_{K_{\mathbf{C}} \in \boldsymbol{K}_{\mathbf{C}}} P(\mathbf{V}(K_{\mathbf{C}})_{\mathbf{B}} \mid \pi(\mathbf{V}(K_{\mathbf{C}}))_{\mathbf{A}}) \prod_{K_{\mathbf{D}} \in \boldsymbol{K}_{\mathbf{D}}} P(\mathbf{V}(K_{\mathbf{D}})_{\mathbf{A}} \mid \pi(\mathbf{V}(K_{\mathbf{D}}))_{\mathbf{A}}) \quad (3.2)$$

Example 3.3

Consider the example in Fig. 3.2 with time points of interest $\mathbf{A}$ and $\mathbf{B}$ and joint intensity matrix Q for continuous-time variables LHT, HA and TROP. As the continuous component has no parents, the joint distribution over time points $\mathbf{B} \setminus \{0\}$ is then given by:

$$P = \prod_{\alpha \in \mathbf{A} \setminus \{0\}} P(DT_{s(\alpha)}) P(BW_{s(\alpha)} \mid BW_\alpha, HF_{s(\alpha)}) P(HF_{s(\alpha)} \mid HF_\alpha, CM_\alpha)$$
$$P(CM_{s(\alpha)} \mid CM_\alpha, DT_\alpha, DT_{s(\alpha)}, LHT_{s(\alpha)}) \prod_{\beta \in \mathbf{B} \setminus \{\max \mathbf{B}\}} \exp(Q(s(\beta) - \beta))$$

The following propositions establish that HTBNs are proper generalizations of both DBNs and CTBNs. For DBNs, the transition graph can be converted to an HTBN graph, where the intra-temporal arcs are replaced by arcs with delay 0 and inter-temporal arcs are replaced by arcs with delay 1.

Proposition 3.3.1. *A DBN $(\mathcal{B}_0, \mathcal{B}_\rightarrow)$, as defined in Definition 2.6, and an HTBN $(\mathcal{B}, G_\mathcal{H}, \Phi, \varnothing)$ define the same joint probability distribution for any set of time points of interest $\mathbf{A}$, if $\mathcal{B}_0 = \mathcal{B}$, $G_\mathcal{H}$ corresponds to the graph of $\mathcal{B}_\rightarrow$, and Φ are the parameters of the DBN.*

Similarly, a CTBN can be interpreted as an HTBN by replacing its arcs by arcs with ϵ delay.

Proposition 3.3.2. *A CTBN $(\mathcal{B}, G_\rightarrow, \mathbf{Q})$, as defined in Definition 2.7, and an HTBN $(\mathcal{B}, G_\mathcal{H}, \varnothing, \mathbf{Q})$ define the same probability distribution for any set of time points of interest $\mathbf{B}$ if $G_\mathcal{H}$ is $G_\rightarrow$ such that each edge is replaced by an edge with delay ϵ.*

For both propositions, the results follow directly from the factorization of CTBNs, DBNs, and HTBNs.

3.4 DISCRETE-TIME CHARACTERIZATION

A natural question is whether the joint distribution defined on an HTBN, given the fixed time points of interest, can also be graphically represented as a regular (discrete-time) Bayesian network. The benefit is that the parameters of the resulting Bayesian network are conditional probabilities, which are easier to understand by domain experts. Furthermore,

this construction is convenient as this implies that we can (dynamically) generate discrete-time versions of the model given time points for which we have observations, and for which we would like to compute marginals. Given this translation, existing algorithms for probabilistic inference in BNs can be employed.

In Section 3.4.1 we describe the procedure to construct a discrete-time graph structure of a hybrid model. The construction for the complete representative BN is given in Section 3.4.2, employing an EM algorithm to obtain the parameters.

3.4.1 *Structural Discretization*

The discretization of a hybrid-time model requires discretizing the continuous components $K_{\mathbf{C}}$ given time points of interest $\mathbf{B}$. Since each continuous component is itself a CTBN, we focus in this subsection mainly on the discretization procedure in terms of CTBNs. The resulting CTBNs are subsequently combined with the discrete components $K_{\mathbf{D}}$ to obtain the discrete version of the HTBN.

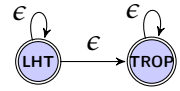
The remainder of this subsection is structured as follows. In Definition 3.5 and Lemma 1 we characterize the independence structure of a CTBN in terms of an infinite set of Bayesian networks which follow the causal structure of the graph. Then, in Definition 3.6 and Theorem 2, we show the existence of a single Bayesian network, called a *representative Bayesian network*, that includes all possible dependences of this set of causal networks, and therefore of the CTBN. In Theorem 3, it is shown how to directly construct the graph of this representative Bayesian network from a given CTBN. Finally, in Definition 3.7, this is tied together with the discrete components of an HTBN.

Definition 3.5 (Associated Causal Graph)**.** *Consider a CTBN with graph* $(G_0, G_\rightarrow)$*, with* $G_0 = (\mathbf{V}(G_0), \mathbf{E}(G_0))$ *and* $G_\rightarrow = (\mathbf{V}(G_\rightarrow), \mathbf{E}(G_\rightarrow))$*, and time-indexed variables* $\mathbf{V}_\mathbf{B}$*. An associated causal graph* $G_\mathbf{B}$*,* $G_\mathbf{B} = (\mathbf{V}(G_\mathbf{B}), \mathbf{E}(G_\mathbf{B}))$*, is the graph with nodes* $\mathbf{V}(G_\mathbf{B}) = \mathbf{V}_\mathbf{B}$ *such that for all* $X, Y \in \mathbf{V}_\mathbf{B}$*:*

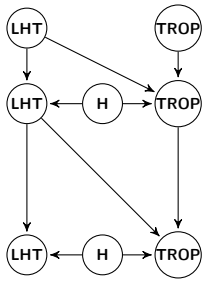
- *If* $X \rightarrow Y \in \mathbf{E}(G_0)$*, then* $X_0 \rightarrow Y_0 \in \mathbf{E}(G_\mathbf{B})$*;*
- *If* $X \rightarrow Y \in \mathbf{E}(G_\rightarrow)$ *and* $\{\beta, s(\beta)\} \subseteq \mathbf{B}$*, then* $X_\beta \rightarrow Y_{s(\beta)} \in \mathbf{E}(G_\mathbf{B})$*.*

and $G_\mathbf{B}$ *does not contain any other arcs.*

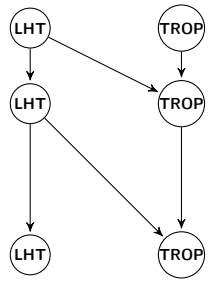
(a) Initial Model G_0



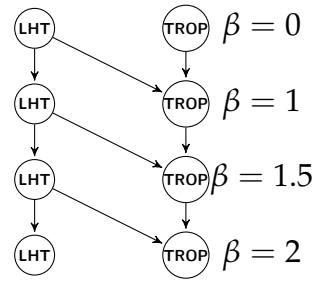
(b) Transition model $G_{\rightarrow}$



(c) A representative BN with $\mathbf{B} = \{0, 1, 2\}$



(d) Causal graph with $\mathbf{B} = \{0, 1, 2\}$



(e) Causal graph with $\mathbf{\Gamma} = \{0, 1, 1.5, 2\}$, $\mathbf{\Gamma} \supseteq \mathbf{B}$

Fig. 3.3: An example of causal graphs and a representative BN for a given CTBN and time points of interest $\mathbf{B}$; H stands for a hidden variable.

Example 3.4

Consider a continuous component with the initial model and transition model shown in Fig. 3.3a and 3.3b. The associated causal graphs of the continuous component with time points of interest $\mathbf{B}$ and $\mathbf{\Gamma}$ are shown in Fig. 3.3d and 3.3e.

In the following lemma, we give a precise characterization of the relationship between CTBNs and causal graphs in terms of independence structures.

Lemma 1. *Let $P(\mathbf{V})$ be the distribution defined by a CTBN with graph $(G_0, G_{\rightarrow})$ and time-indexed variables $\mathbf{V_B}$, and let $X_\beta, Y_{\beta'} \in \mathbf{V_B}$. Then if for all $\mathbf{V_\Gamma}$ with $\mathbf{\Gamma} \supseteq \mathbf{B}$, in the* associated causal graph *$G_\mathbf{\Gamma}$ over $\mathbf{V_\Gamma}$ it holds that $X_\beta \perp\!\!\!\perp_{G_\mathbf{\Gamma}} Y_{\beta'} \mid \mathbf{Z}$, then $X_\beta \perp\!\!\!\perp_{P(\mathbf{V})} Y_{\beta'} \mid \mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V_B} \setminus \{X_\beta, Y_{\beta'}\}$.*

Proof. The data-generating process of CTBNs generates a sequence of events, i.e., observations of random variables indexed by time. Since there is no feedback, i.e., it is a causal process, the joint distribution of these sequences can be described by a causal DAG as given by Definition 3.5 for a set of time points $\mathbf{\Gamma}$ that occur in these sequences. As a result, each dependency $X_\beta \not\perp\!\!\!\perp_{P(\mathbf{V})} Y_{\beta'} \mid \mathbf{Z}$ will be represented by some DAG that includes the variables $\mathbf{V_B}$. □

Next, we define a Bayesian network that represents independences of the CTBN, using the causal graphs from Definition 3.5. Theorem 2 proves that this BN is indeed representative.

Definition 3.6 (Representative Discrete-time Bayesian Network)**.** *Consider a CTBN with graph* $(G_0, G_{\rightarrow})$ *and time-indexed variables* $\mathbf{V_B}$*. A representative discrete-time Bayesian network is a Bayesian network with graph* $G_{\mathbf{B}}$*,* $G_{\mathbf{B}} = (\mathbf{V}(G_{\mathbf{B}}), \mathbf{E}(G_{\mathbf{B}}))$*, which includes at least the set of nodes* $\mathbf{V_B}$ *and given any* $X_\beta, Y_{\beta'} \in \mathbf{V_B}$*, if:*

$$X_\beta \not\perp\!\!\!\perp_{G_{\mathbf{\Gamma}}} Y_{\beta'} \mid \mathbf{Z}$$

for all $\mathbf{Z} \subseteq \mathbf{V_B} \setminus \{X_\beta, Y_{\beta'}\}$ *in some associated causal graph* $G_{\mathbf{\Gamma}}$*,* $\mathbf{\Gamma} \supseteq \mathbf{B}$*, then:*

- $\beta' = s(\beta)$ *implies* $X_\beta \rightarrow Y_{\beta'} \in \mathbf{E}(G_{\mathbf{B}})$*;*
- $\beta = s(\beta')$ *implies* $X_\beta \leftarrow Y_{\beta'} \in \mathbf{E}(G_{\mathbf{B}})$*;*
- $\beta = \beta' \neq 0$ *implies* $H_\beta^{XY} \rightarrow X_\beta \in \mathbf{E}(G_{\mathbf{B}}), H_\beta^{XY} \rightarrow Y_\beta \in \mathbf{E}(G_{\mathbf{B}})$*;*
- $\beta = \beta' = 0$ *and* $X_0 \rightarrow Y_0$ *in all causal graphs implies* $X_0 \rightarrow Y_0 \in \mathbf{E}(G_{\mathbf{B}})$*.*

with H_β^{XY} *a new hidden variable, and* $G_{\mathbf{B}}$ *does not contain any other nodes or arcs.*

Example 3.5

Recall the CTBN example from Fig. 3.3. For the causal graph $G_{\mathbf{B}}$, with time indices $\mathbf{B} = \{0, 1, 2\}$, it holds that $\text{LHT}_2 \perp\!\!\!\perp \text{TROP}_2 \mid \{\text{LHT}_{\mathbf{B}}, \text{TROP}_{\mathbf{B}}\} \setminus \{\text{LHT}_2, \text{TROP}_2\}$, but this independence does not hold for the causal graph $G_{\mathbf{\Gamma}}$, with $\mathbf{\Gamma} = \{0, 1, 1.5, 2\}$. This implies that there exists a possible dependence in the CTBN graph which can not be represented by $G_{\mathbf{B}}$. Since every causal BN contains such independence assumptions that may not hold in the CTBN, which is revealed by increasing the set of time points, it implies there may not exist a causal BN that models all the CTBN's dependences. Therefore, additional variables are introduced in Definition 3.6 which create additional dependences, e.g. a hidden variable with arcs to LHT_1 and TROP_1 in the representative BN as shown in Fig. 3.3c.

Theorem 2. *Let $P(\mathbf{V})$ be the distribution defined in a CTBN with graph $(G_0, G_{\rightarrow})$ and time-indexed variables $\mathbf{V_B}$, and let $X_\beta, Y_{\beta'} \in \mathbf{V_B}$. Then if for the graph of the representative discrete-time Bayesian network $G_\mathbf{B}$ it holds that $X_\beta \perp\!\!\!\perp_{G_\mathbf{B}} Y_{\beta'} \mid \mathbf{Z}$, then $X_\beta \perp\!\!\!\perp_{P(\mathbf{V})} Y_{\beta'} \mid \mathbf{Z}$, with $\mathbf{Z} \subseteq \mathbf{V_B} \setminus \{X_\beta, Y_{\beta'}\}$.*

Proof. Take some dependence $X_\beta \not\perp\!\!\!\perp_{P(\mathbf{V})} Y_{\beta'} \mid \mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V_B} \setminus \{X_\beta, Y_{\beta'}\}$. Because of Lemma 1, we know that this dependence is included in some causal graph associated to the CTBN. Let $\mathbf{\Gamma}_i \supset \mathbf{B}$ be a set of time points such that the associated causal graph $G_{\mathbf{\Gamma}_i}$ represents the ith dependence statement over the set of variables $\mathbf{V_B}$ in the CTBN. Observe that for any $\mathbf{\Gamma}' \supset \mathbf{\Gamma} \supseteq \mathbf{B}$, if $X_\beta \not\perp\!\!\!\perp_{G_\mathbf{\Gamma}} Y_{\beta'} \mid \mathbf{Z}$ then $X_\beta \not\perp\!\!\!\perp_{G_{\mathbf{\Gamma}'}} Y_{\beta'} \mid \mathbf{Z}$. Thus, all the dependences of the CTBN over $V_\mathbf{B}$ will be represented by a causal graph $G_\mathbf{\Gamma}$ such that $\mathbf{\Gamma} = \bigcup_i \mathbf{\Gamma}_i$.

Now consider this causal graph $G_\mathbf{\Gamma}$. It is possible to marginalise out all variables $\mathbf{\Gamma} \setminus \mathbf{B}$ using the procedure described in [78, Definition 4.2.1], which coincides with Definition 3.6 instantiated for the structure of these causal graphs, except that we represent bidirectional arcs in ancestral graphs by means of additional hidden variables. Since $G_\mathbf{\Gamma}$ represents the same independence structure as $G_\mathbf{B}$ ([78, Theorem 4.18]), we obtain $X_\beta \not\perp\!\!\!\perp_{G_\mathbf{B}} Y_{\beta'} \mid \mathbf{Z}$. □

According to Definition 3.6, additional hidden variables $\mathbf{H}$ are introduced for a continuous-time variable C in the associated representative Bayesian network graph when C is d-connected with other continuous-time variables. The introduction of hidden variables makes sure that the states of C at any time point are correlated with the other continuous-time variables when their temporal evolutions are taken into consideration. This models entanglement, a concept that occurs in CTBNs and DBNs [71].

The graphs of representative Bayesian network are defined above by the existence of some causal graph of the CTBN. Clearly, it is not feasible to go over all causal graphs and their independences. Therefore, we provide in the following theorem a simple procedure to directly construct this Bayesian network graph from a given CTBN.

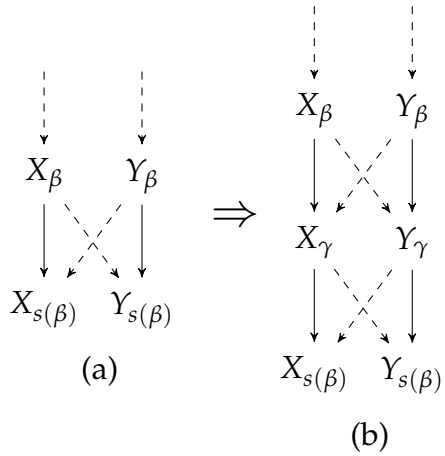
Theorem 3. *Consider a CTBN, with graph $(G_0, G_{\rightarrow})$ and time-indexed variables $\mathbf{V_B}$, and let $G_\mathbf{B}$, $G_\mathbf{B} = (\mathbf{V}(G_\mathbf{B}), \mathbf{E}(G_\mathbf{B}))$, be the graph of the representative discrete-time Bayesian network. Then if $X \rightarrow Y \in G_0$, then $X_0 \rightarrow Y_0 \in \mathbf{E}(G_\mathbf{B})$. Furthermore, for all $X, Y \in \mathbf{V}$, if X and Y are d-connected in $G_{\rightarrow}$ given $\varnothing$, and for all $\beta \in \mathbf{B} \setminus \{0\}$:*

- $H_\beta^{XY} \to X_\beta, H_\beta^{XY} \to Y_\beta \in \mathbf{E}(G_\mathbf{B})$
- *if there is a directed path from X to Y in $G_\to$, then* $X_\beta \to Y_{s(\beta)} \in \mathbf{E}(G_\mathbf{B})$

and $G_\mathbf{B}$ does not contain any other nodes or arcs.

Proof. The nodes and arcs for the initial time slices are obvious because these are exactly included in the causal graphs. Now take some $\beta \in \mathbf{B} \setminus \{0\}$. We will consider the possible relationships between X and Y in the CTBN.

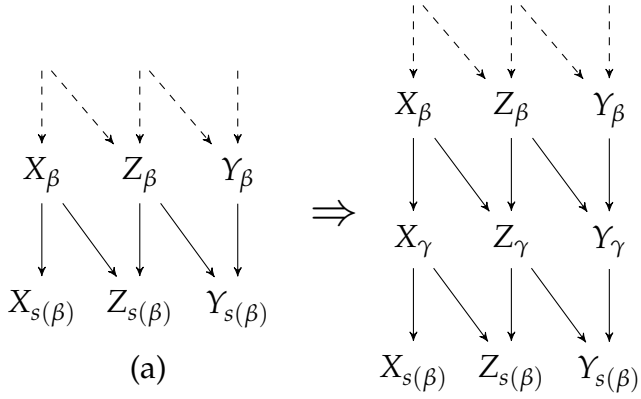
(*i*) Suppose X and Y are d-connected. Then we need to show that there is some causal graph such that X_β is dependent of Y_β for any $\mathbf{Z}$. Consider the case that X and Y are d-connected through some path in the CTBN and consider the causal graph associated to this CTBN in the following two figures:



In (a), $X_{s(\beta)}$ and $Y_{s(\beta)}$ are conditionally independent, e.g., $X_{s(\beta)} \perp\!\!\!\perp Y_{s(\beta)} \mid \{X_\beta, Y_\beta\}$. Now observe that by adding time slices between β and $s(\beta)$, these independences disappear, see (b). Similarly, if the shortest connecting path between X and Y is of length n, any independence between $X_{s(\beta)}$ and $Y_{s(\beta)}$ in a set of variables $\mathbf{V_B}$ will not hold in a causal graph with n additional intermediate time slices between $s(\beta)$ and its predecessor in $\mathbf{B}$. Hence, there will always be a causal graph where there are no independences between $X_{s(\beta)}$ and $Y_{s(\beta)}$ for some conditioning set in $\mathbf{V_B}$.

(ii) Suppose that there is a directed path between X and Y. Then we again need to show that there is some causal graph such that X_β is dependent of $Y_{s(\beta)}$ for any $\mathbf{Z}$. If X and Y are directly connected, then there

will clearly be a dependency of X_β and $Y_{s(\beta)}$, for example a direct dependency in $\mathbf{V_B}$. Now suppose that X and Y are not directly connected. Consider the following figures.

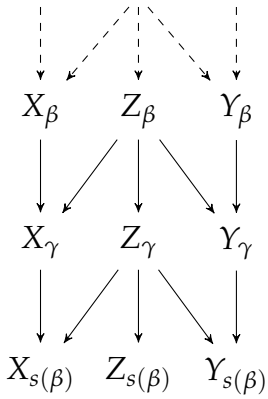


In the causal graph (a) there will be conditional independences between X_β and $Y_{s(\beta)}$, i.e., if we condition on parents of $Y_{s(\beta)}$. Again, by adding intermediate time slices between β and $s(\beta)$, dependences between X_β and $Y_{s(\beta)}$ appear that cannot be blocked by variables in (a), in particular the path $X_\beta \to Z_\gamma \to Y_{s(\beta)}$. Similar to the previous case, directed paths of length n lead to dependences if we add $n-1$ time-slices between β and $s(\beta)$ that cannot be blocked by $\mathbf{V_B}$.

(iii) Suppose X and Y are d-connected, yet, not through a fully directed path. Then we show that in all causal graphs there is a conditional independence between X_β and $Y_{s(\beta)}$. Let $\mathbf{W}$ be the set of variables which are also d-connected to X and Y, i.e., this set includes all variables on paths between X and Y. Then, the claim is that for all causal graph, it holds that:

$$X_\beta \perp\!\!\!\perp Y_{s(\beta)} \mid (\mathbf{W} \setminus \{X, Y\})_\mathbf{B}, Y_\beta$$

Consider some causal graph, for which we know that on all d-connected paths between X and Y there is some divergent node Z (otherwise this path would be fully directed, so case (ii) applies). Then consider the following graph to illustrate why this is true:



Consider that $\beta, s(\beta) \in \mathbf{B}$, but $\gamma \notin \mathbf{B}$. In this case $\mathbf{W} = \{Z\}$, so the conditioning set contains at least $\{Z_\beta, Z_{s(\beta)}, Y_\beta\}$. To d-connect X_β with $Y_{s(\beta)}$ on this path of the CTBN, there should therefore be a path through some $Z_{\gamma'}$, where $\gamma' \notin \mathbf{B}$. Suppose $\gamma' < \beta$, then all paths will be blocked at β, since all the variables at β are in the conditioning set, except for X_β. Now suppose $\beta < \gamma' < s(\beta)$. Then we have the case similar to the figure where $\gamma = \gamma'$. It is clear in this case that such a path would be blocked by X_β because of the v-structure. Finally suppose $\gamma' > s(\beta)$. Then the only way that this opens a path from Y to X is if there is a v-structure on Z, i.e., $X \to Z \leftarrow Y$. But then there is a directed path $X \to Z \to Y$ in the CTBN, so this contradicts the assumption.

(iv) Suppose X and Y are not d-connected. Again, we need show that in all causal graphs there will be at least one conditional independence between X_β and $Y_{s(\beta)}$. This is trivial, because in all graphs $X_\beta \perp\!\!\!\perp Y_{s(\beta)} \mid \varnothing$ because either X and Y are not connected or on all paths between X and Y, there is some v-structure, which then also occurs in the causal graph. The case for X_β and Y_β is analogous.

(v) For any variables X and Y it holds that $X_\beta \perp\!\!\!\perp Y_{s(s(\beta))} \mid \mathbf{V}_{s(\beta)}$. That is, the Markov property holds in CTBNs.

Finally, note that (i)—(v) together imply the claim of the theorem. □

Once we have a discrete version of the continuous components, we are able to discretize a whole HTBN by assembling the components. We use $G_{K_\mathbf{D}}$ to denote the unrolled graph for a discrete-time component $K_\mathbf{D}$ (given time points $\mathbf{A}$). We denote by $G_{K_\mathbf{C}}$ the representative Bayesian network graph of the continuous component $K_\mathbf{C}$ (given time points $\mathbf{B}$). In the following definition, we tie these two discretizations together to discretize the full HTBN. This includes the Bayesian networks associated

to each component, together with a number of edges between these components.

Definition 3.7 (Representative Bayesian Network Structure)**.** *Let $\mathcal{H}$ be an HTBN with continuous components $\mathbf{K_C}$, discrete components $\mathbf{K_D}$, an initial Bayesian network $\mathcal{B} = (G_0, P)$ and a transition model with graph $G_\mathcal{H}$. Given sets of time points $\mathbf{B}$ for continuous-time variables and $\mathbf{A}$ for discrete-time variables, then a representative Bayesian network structure is a graph $G = (\mathbf{V}(G), \mathbf{E}(G))$ where $\mathbf{V}(G) = \bigcup_{K \in \mathbf{K_D} \cup \mathbf{K_C}} \mathbf{V}(G_K)$, $\mathbf{E}(G)$ includes $\bigcup_{K \in \mathbf{K_D} \cup \mathbf{K_C}} \mathbf{E}(G_K)$, and for any continuous-time variable C and discrete-time variable D with $D \in \mathbf{D}, C \in \mathbf{C}$:*

- $C \xrightarrow{0} D \in G_\mathcal{H}$ *implies* $C_\alpha \to D_\alpha \in \mathbf{E}(G)$*, for all* $\alpha \in \mathbf{A}$*;*
- $D \xrightarrow{\epsilon} C \in G_\mathcal{H}$ *implies* $D_a \to C_\beta \in \mathbf{E}(G)$*,* $a = \max\{\alpha \mid \alpha < \beta\}$ *for all* $\beta \in \mathbf{B}$*;*
- $D_0 \xrightarrow{0} C_0 \in G_0$ *implies* $D_0 \to C_0 \in \mathbf{E}(G)$*;*
- $C_0 \xrightarrow{0} D_0 \in G_0$ *implies* $C_0 \to D_0 \in \mathbf{E}(G)$*;*

and G does not contain any other nodes or arcs.

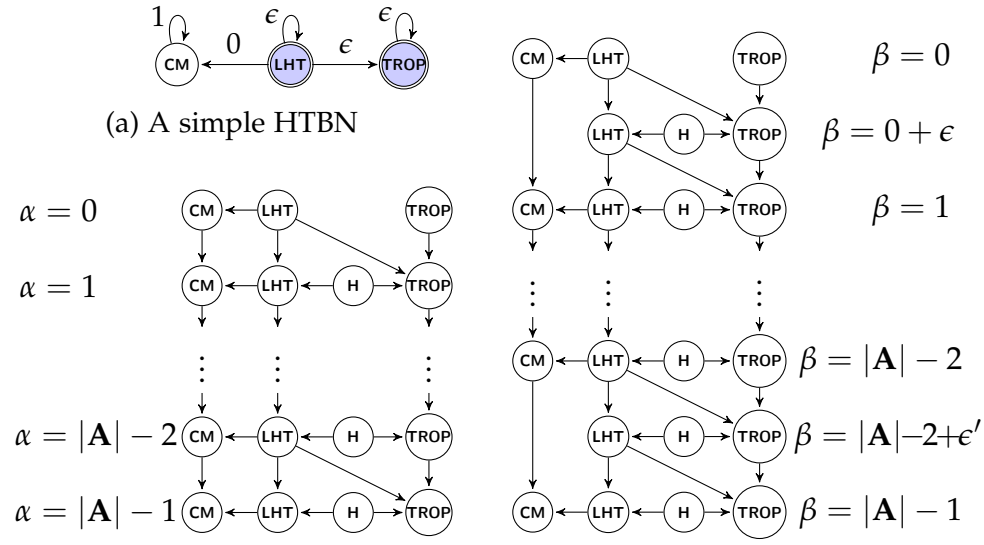
Example 3.6

The graph of the representative Bayesian network for an HTBN is illustrated in Fig. 3.4. Two cases are shown, the graph for $\mathbf{A} = \mathbf{B}$ and $\mathbf{A} \neq \mathbf{B}$.

3.4.2 *Constructing Representative Bayesian Nertworks*

After obtaining the structure of the discrete version of an HTBN, the second step in the procedure of discretization is to estimate the parameters for the representative BN given an HTBN and sets of time points of interest **A**, **B**. The full procedure, using the results above, is given in Algorithm 3.1 and explained below.

The algorithm is based on an Expectation Maximization (EM) procedure [15]. This well-known procedure is traditionally used to learn distributions from data with missing values. In our case, we exploit EM for



(a) A simple HTBN

(b) Same time points of interest $\mathbf{A} = \mathbf{B}$

(c) Different time points of interest: $\mathbf{A} \neq \mathbf{B}$

Fig. 3.4: Discretization of an HTBN.

learning parameters of the representative Bayesian network, as this network includes a number of hidden variables for the continuous components. In order to fit the original joint distribution of a continuous component to a distribution that includes hidden variables, we look upon the hidden variables as if they are missing values. In an ordinary EM procedure parameters of the model are maximized based on expected sufficient statistics, which are determined based on the distribution of observed data and expectations computed at each iteration. In this case, we replace the distribution of observed data by the known distribution given by an HTBN. This is analogous to learning from data since the HTBN parameters are equivalent to the expected sufficient statistics of data generated from the HTBN. As a consequence, this parameter fitting procedure has the same properties as any other EM algorithm, in particular, the procedure results in a local minimum of the Kullback-Leibler divergence between the distributions associated to the HTBN and the representative BN. (see e.g. Section 11.4.7 of [63]).

In the following, the parameters in the BN for continuous-time variables and hidden variables are denoted as $\boldsymbol{\theta}$. We use $\boldsymbol{\theta}^H_{s(\beta)}$ for the prior probability of $H_{s(\beta)}$. Furthermore, we use $\boldsymbol{\theta}^C_{s(\beta)}$ to represent the conditional probability of $C_{s(\beta)}$ conditioned on its parents in the representa-

tive Bayesian network. Finally, to distinguish between the distributions of the hybrid model and representative BN, we use $P_{\mathcal{H}}$ for the distribution of the HTBN and P for the distribution in the BN.

Algorithm 3.1: Construct representative BN

Input: A HTBN $\mathcal{H}$ and sets of time points $\mathbf{A}$, $\mathbf{B}$
Result: Representative BN for $\mathcal{H}$

Construct the representative BN structure according to Def. 3.7
Inherit parameters for discrete-time variables $\mathbf{D}$ from $\mathcal{H}$
Initialize parameters $\boldsymbol{\theta}$ for continuous-time and hidden variables in BN
while $\boldsymbol{\theta}$ *is not converged* **do**
  foreach $s(\beta) \in \mathbf{B}$ **do**
    Compute expectation for $\mathbf{H}_{s(\beta)}$ in BN ▷ E-Step
    $P(\mathbf{H}_{s(\beta)} \mid \mathbf{C}_\beta, \mathbf{C}_{s(\beta)}, \mathbf{D}_a)$
    where $a = \max\{\alpha \mid \alpha < s(\beta), \alpha \in \mathbf{A}\}$
    Let $\Delta = \mathbf{H}_{s(\beta)} \cup \mathbf{C}_\beta \cup \mathbf{C}_{s(\beta)} \cup \mathbf{D}_a$
    foreach $C_{s(\beta)} \in \mathbf{C}_{s(\beta)}$ **do**
      Update parameters for $C_{s(\beta)}$ in BN ▷ M-Step
      $\boldsymbol{\theta}^C_{s(\beta)} \propto \sum_{\Delta \setminus \{\pi(C_{s(\beta)}) \cup \{C_{s(\beta)}\}\}} P(\mathbf{H}_{s(\beta)}, \mathbf{C}_\beta, \mathbf{C}_{s(\beta)}, \mathbf{D}_a)$ where
      $P(\mathbf{H}_{s(\beta)}, \mathbf{C}_\beta, \mathbf{C}_{s(\beta)}, \mathbf{D}_a) = P(\mathbf{H}_{s(\beta)} \mid \mathbf{C}_\beta, \mathbf{C}_{s(\beta)}, \mathbf{D}_a) P_{\mathcal{H}}(\mathbf{C}_\beta, \mathbf{C}_{s(\beta)} \mid \mathbf{D}_a) P_{\mathcal{H}}(\mathbf{D}_a)$
    end
    foreach $H_{s(\beta)} \in \mathbf{H}_{s(\beta)}$ **do**
      Update prior probability for $H_{s(\beta)}$ in BN ▷ M-Step
      $\boldsymbol{\theta}^H_{s(\beta)} = \sum_{\Delta \setminus H_{s(\beta)}} P(\mathbf{H}_{s(\beta)}, \mathbf{C}_\beta, \mathbf{C}_{s(\beta)}, \mathbf{D}_a)$
    end
  end
end
return BN

Algorithm 3.1 outlines the procedure of constructing a representative BN given an HTBN and time points. First, it constructs the structure of the representative BN according to Definition 3.7. In the remainder, it estimates parameters for continuous-time variables given their parents, including the hidden variables. The estimation of parameters for

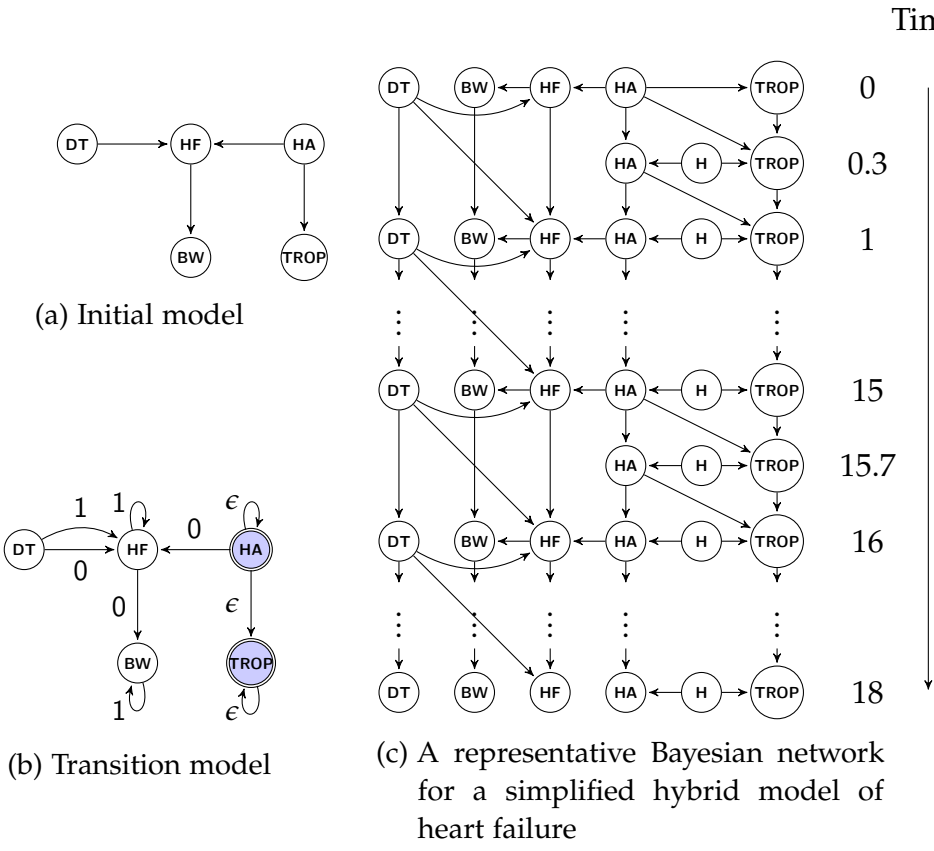
discrete-time variables is straightforward as they can be directly copied from the parameters defined in the HTBN $\mathcal{H}$.

The parameters $\boldsymbol{\theta}^{C}_{s(\beta)}$ and $\boldsymbol{\theta}^{H}_{s(\beta)}$ start with an arbitrary initialization in the BN. In the E-step, we compute the expectation for hidden variables $\mathbf{H}_{s(\beta)}$ given the Markov blanket of $\mathbf{H}_{s(\beta)}$, each of which is represented as a conditional probability as shown on line 7. In the M-step, i.e., from Line 11 onwards, the parameters $\boldsymbol{\theta}^{C}_{s(\beta)}$ and $\boldsymbol{\theta}^{H}_{s(\beta)}$ are updated by marginalizing out variables from Δ by making use of the expectations computed in the E-step and information derived from the HTBN distribution $P_{\mathcal{H}}$. In particular, the distributions $P_{\mathcal{H}}(\mathbf{D}_a)$ and $P_{\mathcal{H}}(\mathbf{C}_\beta, \mathbf{C}_{s(\beta)} \mid \mathbf{D}_a)$ can be computed by marginalisation from the joint distribution of the HTBN as given in Definition 3.4.

3.5 EXPERIMENTS

The power of HTBNs is illustrated in the domain of myocardial contractability in relationship to heart attack, heart failure and its medical treatment, introduced in Section 3.2 and summarized in Fig. 3.2. Of particular interest is the question how the dynamics of the occurrence of heart failure are affected by heart attacks and the administration of digitalis. As discussed in Section 3.2, a single DBN and CTBN can not provide a satisfactory representation of the evolution of variables with different rates: the administering of digitalis or measurement of body weight is regular, in contrast to the more sparse and irregular occurrence of heart attacks.

We now show a simplified HTBN of the heart failure problem given time points of interest **A**, **B**. As the data were unavailable for heart tissue and contractility, the model was simplified by leaving out variables CM and LHT. The graphical representation of the resulting hybrid-time model is shown in Fig. 3.5a and Fig. 3.5b. We parameterized the model partly using medical expert knowledge and partly from data. The unit of time for the transitions is weeks. There are three weekly changing variables in the model, i.e., digitalis (DT), heart failure (HT) and body weight (BW). The parameters of the continuous-time variables (heart attack (HA) and troponin (TROP)) were derived from real data. The transition rates for troponin were parameterized using the MIMIC II Clinical Database [28, 81] which contains thousands of Intensive Care Unit (ICU) patient records gathered from 2001 to 2008. Similarly, some parameters for troponin in the initial model were obtained from the same



(a) Initial model

(b) Transition model

(c) A representative Bayesian network for a simplified hybrid model of heart failure

Fig. 3.5: Simplified hybrid model of heart failure and its associated representative Bayesian network graph given time points of interest.

database. However, due to incompleteness of the clinical information from the MIMIC II Database, we derived the distribution of troponin in the absence of heart attack from the literature [50].

Now we show the procedure to determine the parameters of a representative BN given the HTBN and time points of interest. According to Def. 3.7, a representative BN was generated as shown in Fig. 3.5c, given the sets of time points $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \subset \mathbf{B}$, where $\mathbf{B}$ includes two additional irregular time points for the heart attack events. Taking the parameters in the hybrid model shown in Fig. 3.6 and the constructed BN as input for the EM procedure described in Algorithm 3.1, we computed the parameters of the representative BN. The EM algorithm is implemented in R using the CTBN-RLE reasoning engine [87] and its R interface [99]. Some of the learned parameters are shown in Table 3.1. The parameters for HA at time 0.3 and 1 are different because the time points are

$P(\mathrm{DT}_0 = \mathrm{T})$	0.01
$P(\mathrm{HA}_0 = \mathrm{T})$	0.01
$P(\mathrm{TROP}_0 = \mathrm{N} \mid \mathrm{HA}_0 = \mathrm{T})$	0.15
$P(\mathrm{TROP}_0 = \mathrm{N} \mid \mathrm{HA}_0 = \mathrm{F})$	0.98
$P(\mathrm{HF}_0 = \mathrm{T} \mid \mathrm{DT}_0 = \cdot, \mathrm{HA}_0 = \cdot)$	0.01

DT_t	DT_{t+1}	HF_t	HA_{t+1}	HF_{t+1}
F	F	F	F	0.08
T	F	F	F	0.07
F	T	F	F	0.05
T	T	F	F	0.04
F	F	T	F	0.45
T	F	T	F	0.40
F	T	T	F	0.40
T	T	T	F	0.30
F	F	F	T	0.40
T	F	F	T	0.35
F	T	F	T	0.30
T	T	F	T	0.25
F	F	T	T	0.99
T	F	T	T	0.70
F	T	T	T	0.65
T	T	T	T	0.70

$$Q_{\mathrm{HA}} = \begin{array}{c} \\ F \\ T \end{array} \begin{array}{c} \begin{array}{cc} F & T \end{array} \\ \begin{pmatrix} -0.002 & 0.002 \\ 0.4 & -0.4 \end{pmatrix} \end{array}$$

$$Q_{\mathrm{TROP|HA=T}} = \begin{array}{c} \\ N \\ Abn \end{array} \begin{array}{c} \begin{array}{cc} N & Abn \end{array} \\ \begin{pmatrix} -0.05 & 0.05 \\ 0.02 & -0.02 \end{pmatrix} \end{array}$$

$$Q_{\mathrm{TROP|HA=F}} = \begin{array}{c} \\ N \\ Abn \end{array} \begin{array}{c} \begin{array}{cc} N & Abn \end{array} \\ \begin{pmatrix} -0.01 & 0.01 \\ 0.015 & -0.015 \end{pmatrix} \end{array}$$

Fig. 3.6: Parameters in the hybrid model. Prior probabilities (top) and transition parameters, i.e. $P(\mathrm{HF}_{t+1} = \mathrm{T} \mid \mathrm{DT}_t, \mathrm{DT}_{t+1}, \mathrm{HF}_t, \mathrm{HA}_{t+1})$ for all parent configurations (bottom left) and the intensity matrices for HA and for TROP given HA (bottom right).

Table 3.1: Parameters in the representative BN learned by EM. Parameters for DT, HF are the same as in Fig. 3.6. Prior probabilities for hidden variables H in (a), transition parameters for HA, TROP for all parent configuration, i.e. $P(\mathrm{HA}_{s(\beta)} = \mathrm{T} \mid \mathrm{HA}_\beta, \mathrm{H}_{s(\beta)})$ in (*b*), (*c*), $P(\mathrm{TROP}_{s(\beta)} = \mathrm{Abn} \mid \mathrm{HA}_\beta, \mathrm{TROP}_\beta, \mathrm{H}_{s(\beta)})$ in (*d*), (*e*), $s(\beta) \in \{0.3, 1\}$.

(a)

$\mathrm{H}_{0.3}$	0.51
H_1	0.51

(b)

HA_0	$\mathrm{H}_{0.3}$	$\mathrm{HA}_{0.3}$
F	F	$9 \cdot 10^{-4}$
T	F	0.84
F	T	$2 \cdot 10^{-4}$
T	T	0.94

(c)

$\mathrm{HA}_{0.3}$	H_1	HA_1
F	F	$2 \cdot 10^{-3}$
T	F	0.70
F	T	$5 \cdot 10^{-4}$
T	T	0.81

(d)

HA_0	TROP_0	$\mathrm{H}_{0.3}$	$\mathrm{TROP}_{0.3}$
F	Abn	F	0.99
T	Abn	F	0.99
F	N	F	$5 \cdot 10^{-3}$
T	N	F	0.02
F	Abn	T	0.99
T	Abn	T	0.99
F	N	T	$1 \cdot 10^{-3}$
T	N	T	$8 \cdot 10^{-3}$

(e)

$\mathrm{HA}_{0.3}$	$\mathrm{TROP}_{0.3}$	H_1	TROP_1
F	Abn	F	0.98
T	Abn	F	0.98
F	N	F	0.01
T	N	F	0.04
F	Abn	T	0.99
T	Abn	T	0.99
F	N	T	$3 \cdot 10^{-3}$
T	N	T	0.02

unevenly spaced. In the following, we illustrate the equivalence of the HTBN with the representative BN by computing one of the marginal probabilities, i.e. $P(\mathrm{HA}_{0.3})$.

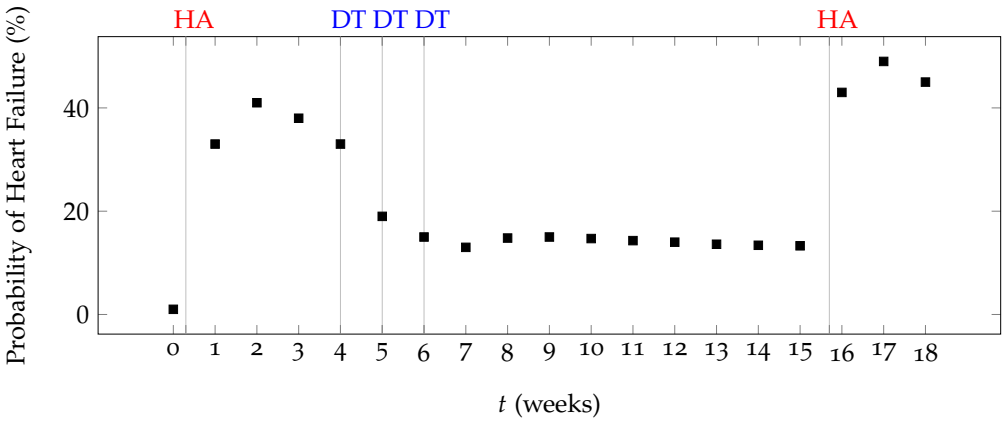


Fig. 3.7: Effects of heart attack and digitalis on heart failure. 'DT' indicates that digitalis was administered at that moment in time. 'HA' indicates that a heart attack was observed. Note that observations for HA are continuous-time, so observed at an arbitrary point in time; digitalis is observed once a week at most.

In the BN, we can compute the marginal of $\mathrm{HA}_{0.3}$ by:

$$\begin{aligned} P(\mathrm{HA}_{0.3} = T) &= \sum_{\mathrm{HA}_0, \mathrm{H}_{0.3}} P(\mathrm{HA}_0, \mathrm{H}_{0.3}, \mathrm{HA}_{0.3} = T) \\ &= \sum_{\mathrm{HA}_0, \mathrm{H}_{0.3}} P(\mathrm{HA}_{0.3} = T \mid \mathrm{HA}_0, \mathrm{H}_{0.3}) P(\mathrm{HA}_0) P(\mathrm{H}_{0.3}) \\ &= 9 \cdot 10^{-4} \cdot 0.99 \cdot 0.49 + 0.84 \cdot 0.01 \cdot 0.49 \\ &\quad + 2 \cdot 10^{-4} \cdot 0.99 \cdot 0.51 + 0.94 \cdot 0.01 \cdot 0.51 \\ &\approx 0.009 \end{aligned}$$

As HA has no parent in the HTBN model, we can directly compute its distribution at time 0.3 using the intensity matrix and initial distribution of HA described in Fig. 3.6:

$$\begin{aligned} \mathbf{P}(\mathrm{HA}_{0.3}) &= \mathbf{P}(\mathrm{HA}_0) \exp(Q_{\mathrm{HA}} \cdot 0.3) \\ &= [0.99, 0.01] \begin{pmatrix} 0.9994348 & 0.0005652 \\ 0.1130463 & 0.8869537 \end{pmatrix} \\ &\approx [0.991, 0.009] \end{aligned}$$

with probability row vectors $\mathbf{P}(\mathrm{HA}_{0.3}) = [P(\mathrm{HA}_{0.3} = F), P(\mathrm{HA}_{0.3} = T)]$, and $\mathbf{P}(\mathrm{HA}_0) = [P(\mathrm{HA}_0 = F), P(\mathrm{HA}_0 = T)]$.

Finally, to show the behavior of the model we computed the probability distribution of heart failure for a period of 19 weeks given the

observed (regular or irregular) evidence. Results of this experiment are plotted in Fig. 3.7. The plot shows the negative effects of a heart attack (see the jumps at time $t = 1$, $t = 2$ and $t = 16$) and the positive effect of digitalis on heart failure (see the rapid fall at time $t = 5$). The model also implies that the condition of the heart stabilizes after administering the drug through an increase in the contractility. However, a damaged heart does not fully recover, not even with the help of digitalis.

3.6 DISCUSSION

We have described hybrid time Bayesian networks as a means to model dynamic systems. In particular, HTBNs are suitable when a combination of regular and irregularly changing variables best describe the domain or when multiple time granularities need to be modeled. The proposed approach generalizes both continuous-time and discrete-time Bayesian networks. HTBNs allow reasoning over irregularly spaced evidence, a property inherited from CTBNs, as well as regular time sliced evidence, derived from DBNs. This is a significant contribution in terms of ease of modeling complex dynamic systems. Each variable can be modeled using the appropriate mechanism, which is particularly helpful when constructing models from expert knowledge. Furthermore, we established a mapping of hybrid-time networks into a standard BN given a set of time points of interest. The inference problem in HTBNs is therefore reduced to a problem for which efficient solutions exist.

DBNs were proposed by Dean and Kanazawa as early as 1989 [14] and have been extensively studied and applied since then, notably by Murphy [64]. CTBNs [71] are of a more recent date, but have seen various applications over the last decade. Applications range from detecting attacks in computer networks [102], analyzing social network dynamics [22] and modeling heart failure [26]. Note that in this chapter we do not aim to improve upon the medical analysis provided in the work by Gatti et al. [26], but instead studied the heart failure domain to show an application of HTBNs in a relevant use case.

There is also some relation between non-stationary dynamic Bayesian networks (nsDBN) and the formalism proposed here. In nsDBNs the structure and parameters change at particular time points [35, 80]. The approaches are related in the sense that non-stationary Bayesian networks allow for different time granularities of the (complete) temporal

process. The key difference here is that we consider the case where different random variables evolve at different rates.

A limitation of HTBNs is that so far the granularities of discrete-time variables are assumed to be fixed, as the focus of this chapter has been on the combination of continuous and discrete-time models. A future extension of the work presented here would be combining different discrete-time granularities within the hybrid-time framework. This is related to the work on irregular-time Bayesian networks (ITBNs) [75] and to the work by van der Heijden and Lucas [40].

There is also some possible future work on inference. Currently, inference in HTBNs is limited by the exponential increase of dependencies when discretizing large continuous components, as currently all variables are connected. It appears possible to optimize the BN construction to only take into account dependencies that have significant influence on the temporal process. A more efficient, approximate structure would alleviate the complexity problem. Note that CTBNs also suffer from this problem, preventing exact inference in large networks.

Since the BN construction presented here may become infeasible for large networks with many time points of interest, another direction for further study would be inference methods that operate directly on the HTBN instead of constructing a BN first. Possibilities include a sampling approach to inference as well as expectation propagation, which has been proposed for inference in CTBNs [73].

Finally, it remains of interest to study learning HTBNs from data. We expect that a general approach to parameter learning can be constructed based on existing work on parameter learning in BNs and CTBNs. Yet, it will require some extensions to take the interactions between discrete and continuous components into account. Structure learning from data is for now an open problem. Nevertheless, hand-crafted HTBN models can be applied already using the inference approach developed in this chapter.

4

LEARNING PARAMETERS OF HYBRID TIME BAYESIAN NETWORKS

In this chapter, we discuss the problem of estimating parameters for a hybrid time Bayesian network described in the previous chapter. We consider the parameter estimation in HTBNs where data can be complete or incomplete. For time series, *complete data* is a set of one or more full trajectories describing the state changes of a variable over time. In other words, we know all the state transitions, i.e., what is the current and the next state for the variable, and exactly when such a transition occurs. However, complete data is rarely available for many real-world applications. To some extent, the data almost always are incomplete or contain missing values, partly due to technical difficulties or measurement errors. For time series, the incompleteness can also be attributed to the fact that we may know the states at two time points, but the number of transitions and states between these time points are unknown in the data. This is even more likely for continuous-time variables because of the continuous nature of time. In this chapter, thus, we are devoting ourselves to discussing the learning task in HTBNs from incomplete data where only continuous-time variables are partially observed while discrete-time variables are fully observed.

In the case of complete data, we use a closed-form solution of *maximum a posteriori* (MAP) estimation to determine parameters for an HTBN by exploiting the MAP for its discrete and continuous components. In the case of incomplete data, HTBNs are described by the probabilistic modeling language *Stan*, which allows us to use an existing MCMC method to estimate the posterior distribution of the parameters in HTBNs. We also present numerical experimental results of learning parameters for HTBNs both from complete and incomplete data.

4.1 THE HEART FAILURE EXAMPLE

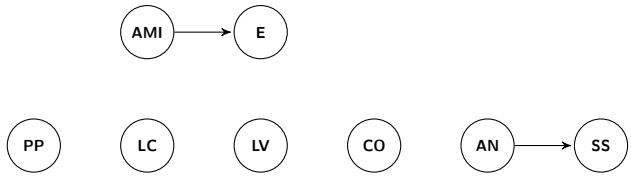
To provide an illustrative learning problem of HTBNs, we start with a medical example that models the clinical condition of heart failure. Modeling the dynamics of heart failure has been the focus of several earlier studies as a temporal modeling problem. For example, Gatti *et, al.* [26] made an attempt to predict the likely evolution of heart failure by fully exploiting CTBNs.

Example 4.1

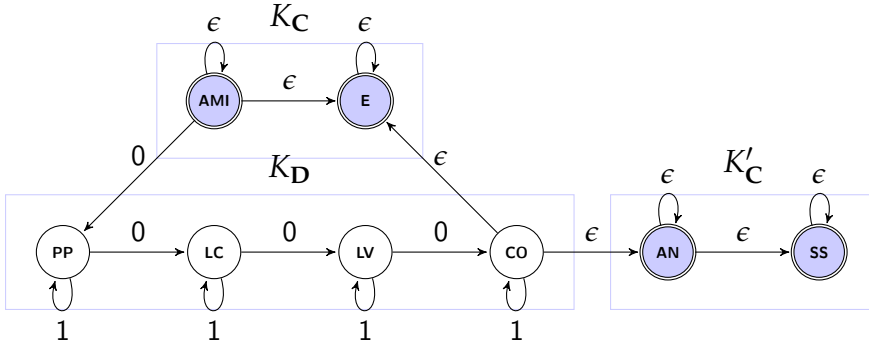
Heart failure is said to be cardiogenic when the cardiac muscle is the organ from which the circulatory failure results. The strength of the heart muscle determines the heart's function as a pump (PP). Cardiogenic heart failure may be caused by acute myocardial infarction (AMI). An AMI reduces blood 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. 4.1, the complete hybrid time Bayesian network $\mathcal{H}$ is shown.

As shown in Fig. 4.1b, there is only one discrete component $K_{\mathbf{D}}$ over four discrete-time variables $V(K_{\mathbf{D}}) = \{\text{PP}, \text{LC}, \text{LV}, \text{CO}\}$ with one continuous-time variable as parent $\pi(V(K_{\mathbf{D}})) = \{\text{AMI}\}$. In addition, we have two continuous components, namely $K_{\mathbf{C}}$ over two continuous-time variables $V(K_{\mathbf{C}}) = \{\text{AMI}, \text{E}\}$ and $K'_{\mathbf{C}}$ over variables $V(K'_{\mathbf{C}}) = \{\text{AN}, \text{SS}\}$. The two continuous components share a single variable CO as their parent, i.e., $\pi(V(K_{\mathbf{C}})) = \pi(V(K'_{\mathbf{C}})) = \{\text{CO}\}$.

For the heart failure model as shown in Fig. 4.1, given time points of interest $\mathbf{A}$ for discrete component $K_{\mathbf{D}}$ and $\mathbf{B}$ for continuous components $K_{\mathbf{C}}$ and $K'_{\mathbf{C}}$, $\mathbf{A} \subseteq \mathbf{B}$, and parameters including conditional probabilities for all discrete-time variables and joint intensity matrix Q and Q' for continuous components $K_{\mathbf{C}}$ and $K'_{\mathbf{C}}$ respectively, the joint distribution P



(a) Initial model



(b) Transition model

Fig. 4.1: 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. 4.1b, 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. 4.1b.

over time points excluding the starting time point $\mathbf{B} \setminus \{0\}$ is then given by:

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

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

With three components over eight variables, the model of heart failure is considered to be a representative example to illustrate some important issues regarding learning parameters of HTBNs. The model has a reasonable complexity in terms of the number of components, namely two continuous components with each having two variables and one discrete component over four variables. In addition, it reveals an important learning issue in HTBNs in practice, i.e., the learning performance when the HTBN contains a cycle between components. In this case, such an issue is demonstrated in the heart failure model by the continuous component $K_{\mathbf{C}}$ and the discrete component $K_{\mathbf{D}}$ forming a cycle in the graph.

4.2 RELATED WORK

There has been some work dealing with parameter estimation for DBNs and CTBNs from incomplete data. Expectation Maximization (EM), originally presented by Dempster *et al.* (1977) [15], is a widely used and accepted algorithm to deal with incomplete data by optimizing a likelihood function. The EM algorithm is a general iterative procedure with each iteration alternating between an expectation (E) step and a maximization (M) step. In the E-step, the algorithm uses the current parameters to compute the expected sufficient statistics and in the M-step, it derives a new set of parameters by performing maximum likelihood estimation. A comprehensive reading of its history, derivation and applications in a single textbook can be found in the work by McLachlan *et al.* (1997) [61].

The EM algorithm is often also used for optimizing likelihood functions in cases where models are specified partly observed and partly

unobserved. One paradigm of these models are hidden Markov models (HMMs) which contain two groups of components: the observed components called observations, and the unobserved component called hidden states. In HMMs, an ad-hoc learning algorithm is derived from the general EM algorithm, known as the *forward-backward* or *Baum-Welch* algorithm, for which a comprehensive explanation can be found in the work by Jurafsky *et al.* [13]. This algorithm allows us to perform efficient inference by making use of dynamic programming in two passes. By definition, the first pass goes forward in time while the second pass goes backward in time.

The EM algorithm is further extended to more general probabilistic graphic models, such as Bayesian networks, which is extensively explained by Koller and Friedman [49]. As the algorithm is not restricted to parameter estimation only, it has also been adapted for model selection to search for an optimal network structure [23, 24], known as *structural EM* and abbreviated as *SEM*. In addition, the SEM algorithm has been recently extended to two variants of temporal Bayesian networks, the dynamic Bayesian networks (DBNs) [67] and the continuous time Bayesian networks (CTBNs) [73], which also model the evolution of a set of random variables over time. However, the sufficient statistic in each iteration differs between CTBNs and DBNs. Besides the expected number of transitions from one state to another, the sufficient statistic in CTBNs also consists of the expected amount of time for a variable to stay in a state.

In theory, the EM approach could be applied to learn 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 difficult because of the available method for inference in HTBNs. The current approach proposed by [55] 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 expensive, which is feasible if the translation has to occur only once. In the EM procedure, however, this translation has to be done for every iteration, which makes it an impractical approach for parameter learning.

Another idea for addressing the problem of estimating parameters from incomplete data in HTBNs is the use of *Markov chain Monte Carlo* (MCMC) methods [27, 37]. MCMC is a large class of sampling algorithms, the essence of which is the construction of a homogeneous

Markov chain with a *stationary distribution*, also known as *equilibrium distribution*. The algorithm generates a sequence of independent and identically distributed samples, for the space on which the desired posterior distribution is defined. Such a general approach plays a significant role in statistics, econometrics, physics and computing science over the last three decades. The popularity of the MCMC algorithm is at least partly due to the fact that the algorithm provides an effective solution within reasonable time for hard learning and inference problems. In the context of Bayesian networks, the MCMC algorithm is widely used for structure learning [18, 25, 34, 79, 93] and parameter estimating [17].

4.3 PARAMETER ESTIMATION IN HTBNS

In this section, we discuss the problem of estimating parameters for hybrid time Bayesian networks from training data. We assume that the network structure is known prior to parameter learning. We address the parameter learning in two general cases. In the first case, the training data is *complete*, i.e., all variables in a given network are fully observed. In other words, we have all the information of a transition, i.e., what is the current and next state for a variable and when such a transition occurs. In the second case, we consider the training data as *incomplete*, or *partial*, as the values of some variables at some time points are missing. More precisely, our focus in this chapter is on continuous-time variables which can be partially observed. This is based on the intuition that continuous-time variables are more likely to be *only* observed at some time points and to be missing for the rest of the time points.

A central concept of parameter learning is the likelihood function that measures the probability for training data. The likelihood function is computed by assigning a probability to each instantiation of random variables in the training data. It therefore measures the effect of the choice of parameters on the training data in terms of probability. Here we focus on the basic principles behind the likelihood function in hybrid time Bayesian networks.

In hybrid time Bayesian networks, there are two types of parameters we need to estimate from training data: conditional probability tables (CPTs) for discrete-time variables and intensity matrices for continuous-time variables. Throughout this section, we make use of existing results on *maximum likelihood estimation* and *maximum a posteriori (MAP)* in DBNs [64] and CTBNs [72] in order to learn parameters from com-

plete data, as HTBNs can be simply viewed as a combination of DBNs and CTBNs. In the next subsection, we will first discuss the likelihood function for HTBNs, which can be decomposed into those for DBNs and CTBNs. In the end of this section, we will address the issue of computing log-likelihood when training data is incomplete. In particular, we will demonstrate that the log-likelihood can not be computed in closed form from incomplete training data.

4.3.1 *Likelihood of Complete Data*

The *likelihood* for a hybrid time Bayesian network given a choice of parameters is the probability of each instantiation in the data. Given a choice of parameters, we assume that each instantiation in the data is simply a random sample from the model, i.e., the instantiations are independent and identically distributed (IID). Consider we have an HTBN $\mathcal{H}$ with a set of parameters $\boldsymbol{\theta}$ (a set of conditional intensity matrices and CPTs), and a partition over a set of continuous components $\mathbf{K_C}$ and a set of discrete components $\mathbf{K_D}$. Given training data $\mathcal{D}$ describing complete trajectories over random variables in the model $\mathcal{H}$, the log-likelihood function for the HTBN $\mathcal{H}$ can be decomposed into its continuous and discrete components:

$$\ell_{\mathcal{H}}(\boldsymbol{\theta} : \mathcal{D}) = \sum_{K_C \in \mathbf{K_C}} \ell_{K_C}(\boldsymbol{\theta}_{K_C} : \mathcal{D}) + \sum_{K_D \in \mathbf{K_D}} \ell_{K_D}(\boldsymbol{\theta}_{K_D} : \mathcal{D}) \tag{4.1}$$

where $\boldsymbol{\theta}_{K_C}$ are a set of parameters for a continuous component K_C and $\boldsymbol{\theta}_{K_D}$ for a discrete component K_D. In Equation 4.1, the log-likelihood function for the HTBN, denoted by $\ell_{\mathcal{H}}(\boldsymbol{\theta} : \mathcal{D})$, is decomposed into the log-likelihood function for each continuous component, indicated by $\ell_{K_C}(\boldsymbol{\theta}_{K_C} : \mathcal{D})$, and for each discrete-time component by $\ell_{K_D}(\boldsymbol{\theta}_{K_D} : \mathcal{D})$.

The likelihood of each component of an HTBN is similar to the likelihood in CTBNs and DBNs. However, the variables in a component should also be taken into consideration when they are parents for the other components. To complete the likelihood function for an HTBN, we need to define the likelihood functions for both the discrete and continuous components.

LIKELIHOOD OF DISCRETE-TIME COMPONENTS Given a choice of parameters $\boldsymbol{\theta}_{K_D}$ for a discrete component K_D, the log-likelihood for training data $\boldsymbol{\mathcal{D}}$ is given by:

$$\ell_{K_D}(\boldsymbol{\theta}_{K_D} : \boldsymbol{\mathcal{D}}) = \sum_{D \in V(K_D)} (\ln P(D[0] \mid \pi(D[0]) : \boldsymbol{\theta}_{K_D}) + \sum_{\alpha \in \mathbf{A} \setminus \{0\}} \ln P(D[\alpha] \mid \pi(D[\alpha]) : \boldsymbol{\theta}_{K_D})) \tag{4.2}$$

where $V(K_D)$ are variables in the component K_D, $D[0]$ and $D[\alpha]$ are assignments to discrete-time variable D at time 0 and α in the training data $\boldsymbol{\mathcal{D}}$ respectively, $\pi(D[0])$ and $\pi(D[\alpha])$ are the assignments to the parents of variable D in the initial model and in the transition model, respectively.

When variables are fully observed over time, the computation of the likelihood can be summarized in terms of sufficient statistics. To put it simply, a sufficient statistic is a function of training data that summarizes the relevant information for computing the likelihood. The sufficient statistic is computed by counting the number of instantiation of variables in a particular state in the training data. For a discrete-time variable D, the sufficient statistics consist of two parts, one for the initial model and one for the transition model. For the former, the sufficient statistic is indicated by $M[\mathbf{u}, d]$, the number of counts that variable D takes the value d and its parents take the values $\mathbf{u}$ at time 0 in the training data. Accordingly, the sufficient statistic is indicated by $M[\mathbf{u}', d]$ in the latter, which is the count of instantiation of $D[\alpha] = d$ and $\pi(D[\alpha]) = \mathbf{u}'$ for time points after time 0 in the data, i.e., α, $\alpha \in \mathbf{A} \setminus \{0\}$.

We use the notation $\theta_{K_D}^{d|\mathbf{u}}$ to represent a parameter for discrete-time variable D in the component K_D where D takes a value d and its parents in the initial model take values $\mathbf{u}$. The notation $\theta_{K_D}^{d|\mathbf{u}'}$ is defined analogously in the transition model. Together with the sufficient statistics $M[\mathbf{u}, d]$ and $M[\mathbf{u}', d]$ given in the previous paragraph, an alternative representation for the log-likelihood function of Equation 4.2 can be given as follows:

$$\ell_{K_D}(\boldsymbol{\theta}_{K_D} : \boldsymbol{\mathcal{D}}) = \sum_{D \in V(K_D)} (\sum_{\mathbf{u}} \sum_{d} M[\mathbf{u}, d] \ln \theta_{K_D}^{d|\mathbf{u}} + \sum_{\mathbf{u}'} \sum_{d} M[\mathbf{u}', d] \ln \theta_{K_D}^{d|\mathbf{u}'}) \tag{4.3}$$

LIKELIHOOD OF CONTINUOUS-TIME COMPONENTS Similarly, the log-likelihood for a continuous component K_C can also be represented

in terms of individual variables $C \in V(K_C)$, where $V(K_C)$ are all variables in the component K_C. However, parameters for a continuous-time variable C consist of two parts: $\mathbf{q}_{K_C}^C$ represents *when* variable C's next transition will occur and $\boldsymbol{\theta}_{K_C}^C$ models *where* variable C will transition, that is, to what state. Given a choice of parameter $\boldsymbol{\theta}_{K_C}$ and $\mathbf{q}_{K_C}$ for a continuous component K_C, the log-likelihood function $\ell_{K_C}(\boldsymbol{\theta}_{K_C}, \mathbf{q}_{K_C} : \boldsymbol{\mathcal{D}})$ for continuous component C is decomposed into those for each variable in the component, as given by:

$$\begin{aligned}\ell_{K_C}(\boldsymbol{\theta}_{K_C}, \mathbf{q}_{K_C} : \boldsymbol{\mathcal{D}}) &= \sum_{C \in V(K_C)} \ell_C(\mathbf{q}_{K_C}^{C|\pi(C)}, \boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) \\ &= \sum_{C \in V(K_C)} \ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) + \sum_{C \in V(K_C)} \ell_C(\boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) \quad (4.4)\end{aligned}$$

where $\mathbf{q}_{K_C}^{C|\pi(C)}$ and $\boldsymbol{\theta}_{K_C}^{C|\pi(C)}$ are parameters for a continuous-time variable C in the component K_C given its parents $\pi(C)$. Parents $\pi(C)$ may include both continuous-time and discrete-time variables.

The likelihood of training data $\boldsymbol{\mathcal{D}}$ described in Equation 4.4 can be further described by two terms: the term $\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}})$ describing the probability of the sequence of durations (i.e. time spend in a state) and the term $\ell_C(\boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}})$ describing the probability of the sequence of state transitions. As all the transitions are observed, we can compute sufficient statistics $T[c|\mathbf{u}]$, the amount of time C spends in state c with its parents taking values $\mathbf{u}$, and $M[c, c']$, the number of times C transitions from state c to state c' for $c \neq c'$. Now we can rewrite the term $\ell_C(\boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}})$ in terms of sufficient statistics $M[c, c']$:

$$\ell_C(\boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) = \sum_{\mathbf{u}} \sum_{c} \sum_{c' \neq c} M[c, c' \mid \mathbf{u}] \ln \theta_{K_C}^{cc'|\mathbf{u}} \quad (4.5)$$

It is important to keep in mind that the term $T[c \mid \mathbf{u}]$ can be decomposed into two different kinds of durations: the total amount of time before a transition in variable C and the total amount of time before a transition in one of the parents of variable C. In the following, we make a distinction between these two types of transitions.

Let $r = \langle c_r, t_r, c_r' \rangle \in \boldsymbol{\mathcal{D}}$ be the transition where variable C transitions to state c_r' after spending the amount of time t_r in state c_r. When one of the parents of variable C in the transition r changes while C stays in the same state, i.e., $c_r = c_r'$, variable C stays in the state for *at least* the

duration of time t_r and the probability for the duration in the transition r is given as below:

$$\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : r) = -q_{K_C}^{c|\mathbf{u}} \cdot t_r \tag{4.6}$$

Otherwise, variable C changes in the transition r, i.e., $c_r \neq c'_r$, the probability for variable C staying for the duration t_r is:

$$\begin{aligned}\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : r) &= \ln(q_{K_C}^{c|\mathbf{u}} \exp(-q_{K_C}^{c|\mathbf{u}} \cdot t_r)) \\ &= -q_{K_C}^{c|\mathbf{u}} \cdot t_r + \ln q_{K_C}^{c|\mathbf{u}}\end{aligned} \tag{4.7}$$

Now we can decompose the likelihood $\ell_C(\boldsymbol{q}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}})$ as a sum of the likelihood for each individual transition $\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : r)$, where the value of variable C does or does not change, given by Equation 4.6 and 4.7, respectively. Together with sufficient statistics $T[c|\mathbf{u}]$, i.e., the amount of time variable C stays at state c and $\pi(C) = \mathbf{u}$, computed by summing up the duration in each transition indicated by t_r in Equation 4.6-4.7, and $M[c \mid \mathbf{u}] = \sum_{c'} M[c, c' \mid \mathbf{u}]$, the total number of transitions for variable C leaving state c (implicitly indicated by 0 in Equation 4.6 and by 1 in Equation 4.7), the term $\ell_C(\boldsymbol{q}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}})$ can be rewritten as:

$$\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) = \sum_{\mathbf{u}} \sum_{c} (M[c \mid \mathbf{u}] \ln q_{K_C}^{c|\mathbf{u}} - q_{K_C}^{c|\mathbf{u}} \cdot T[c \mid \mathbf{u}]) \tag{4.8}$$

Combined with Equation 4.5-4.8, the log-likelihood for variable C in the component K_C can be computed by summing out the likelihood for all state transitions and time durations, formally:

$$\begin{aligned}\ell_C(\mathbf{q}_{K_C}^{C|\pi(C)}, \boldsymbol{\theta}_{K_C}^{C|\pi(C)} : \boldsymbol{\mathcal{D}}) = &\sum_{\mathbf{u}} \sum_{c} M[c \mid \mathbf{u}] \ln q_{K_C}^{c|\mathbf{u}} - q_{K_C}^{c|\mathbf{u}} \cdot T[c \mid \mathbf{u}] \\ &+ \sum_{\mathbf{u}} \sum_{c} \sum_{c' \neq c} M[c, c' \mid \mathbf{u}] \ln \theta_{K_C}^{cc'|\mathbf{u}}\end{aligned} \tag{4.9}$$

4.3.2 *MAP Estimates with Complete Data*

Maximum a posterior (MAP) is used to search for parameters that maximize the posterior probability, written as:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln P(\boldsymbol{\theta} \mid \boldsymbol{\mathcal{D}}) \tag{4.10}$$

By applying the Bayes' rule, the posterior probability in Equation 4.10 can be computed from its log-likelihood and the prior for parameters $\boldsymbol{\theta}$:

$$\begin{aligned}\operatorname*{argmax}_{\boldsymbol{\theta}} \ln P(\boldsymbol{\theta} \mid \boldsymbol{\mathcal{D}}) &= \operatorname*{argmax}_{\boldsymbol{\theta}} \ln \left(\frac{P(\boldsymbol{\mathcal{D}} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\boldsymbol{\mathcal{D}})} \right) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \ln P(\boldsymbol{\mathcal{D}} \mid \boldsymbol{\theta}) + \ln P(\boldsymbol{\theta}) \end{aligned} \tag{4.11}$$

According to Equation 4.11, $\hat{\boldsymbol{\theta}}$ is the maximum of a function that sums together the log-likelihood and the probability of parameters $\boldsymbol{\theta}$ taking a value. Therefore, it biases the parameter estimation away from undesirable parameter values, such as those conditional probabilities involving zeros when the training data is relatively small.

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 $\boldsymbol{\theta}^{d|\mathbf{u}}_{K_D}$, an appropriate conjugate prior is a Dirichlet distribution, where

$$\boldsymbol{\theta}^{d|\mathbf{u}}_{K_D} \sim \mathrm{Dir}(\gamma_1, \ldots, \gamma_k)$$

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

$$\boldsymbol{\theta}^{d|\mathbf{u}}_{K_D} \mid \boldsymbol{\mathcal{D}} \sim \mathrm{Dir}(\gamma_1 + M[d_1 \mid \boldsymbol{u}], \ldots, \gamma_k + M[d_k \mid \boldsymbol{u}])$$

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:

$$\hat{\theta}^{d|\mathbf{u}}_{K_D} = \frac{\gamma_{d|\mathbf{u}} + M[\mathbf{u}, d]}{\gamma_{\mathbf{u}} + M[\mathbf{u}]} \qquad \hat{\theta}^{d|\mathbf{u}'}_{K_D} = \frac{\gamma_{d|\mathbf{u}'} + M[\mathbf{u}', d]}{\gamma_{\mathbf{u}'} + M[\mathbf{u}']} \tag{4.12}$$

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 $\boldsymbol{\theta}^{c|\mathbf{u}}_{K_C}$ is the Dirichlet distribution $\boldsymbol{\theta}^{c|\mathbf{u}}_{K_C} \sim \mathrm{Dir}(\mu^{cc_1|\mathbf{u}}, \ldots, \mu^{cc_n|\mathbf{u}})$. For the exponential parameter $q_{K_\mathbf{C}}$ the conjugate prior is the Gamma

distribution, $q_{K_C} \sim \Gamma(\mu^{\mathbf{u}}, \eta^{\mathbf{u}})$. We can then derive the MAP for a continuous component:

$$\hat{q}^{c|\mathbf{u}}_{K_C} = \frac{\mu^{c|\mathbf{u}} + M[c \mid \mathbf{u}]}{\eta^{c|\mathbf{u}} + T[c \mid \mathbf{u}]} \qquad \hat{\theta}^{cc'|\mathbf{u}}_{K_C} = \frac{\mu^{cc'|\mathbf{u}} + M[c, c' \mid \mathbf{u}]}{\mu^{c|\mathbf{u}} + M[c \mid \mathbf{u}]} \tag{4.13}$$

where

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

4.3.3 *Incomplete Data*

In this subsection, we address the problem of computing log-likelihood for an HTBN given incomplete data $\mathcal{D}$, where some instantiations of all of the variables in the HTBN corresponding to given time points $\mathbf{B}$ are missing in the data $\mathcal{D}$. By applying the sum rule of probability, we can compute the likelihood for the HTBN given partial trajectory $\mathcal{D}$ by summing out the missing instantiations relative to the time points $\mathbf{B}$ from the likelihood for the instantiations over the time points $\mathbf{B}$.

More precisely, given time points $\mathbf{B}$ for an HTBN $\mathcal{H}$ with the corresponding graph G, a set of parameters $\boldsymbol{\theta}$, and its associated joint distribution $P(V(G)_{\mathbf{B}})$ as defined by Eq. 3.2 in the preceding chapter, we can write the log-likelihood of the model $\mathcal{H}$ given the partial trajectory $\mathcal{D}$ in terms of the likelihood over $\mathbf{B}$ in the following:

$$\ell_{\mathcal{H}}(\boldsymbol{\theta} : \mathcal{D}) = \ln \sum_{V(G)_{\mathbf{B}} \setminus \xi(V(G))} P(V(G)_{\mathbf{B}}) \tag{4.14}$$

where $\xi(V(G))$ are the assignments to the variables $V(G)$ in the data $\mathcal{D}$.

In the following, we are assuming the typical situation where continuous-time random variables are observed at arbitrary points in time. This also means that almost everywhere else they are unobserved. Since other variables are typically directly related to the values of continuous-time random variables on discrete-time points (by the definition of 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 a closed form. This is illustrated in the following examples.

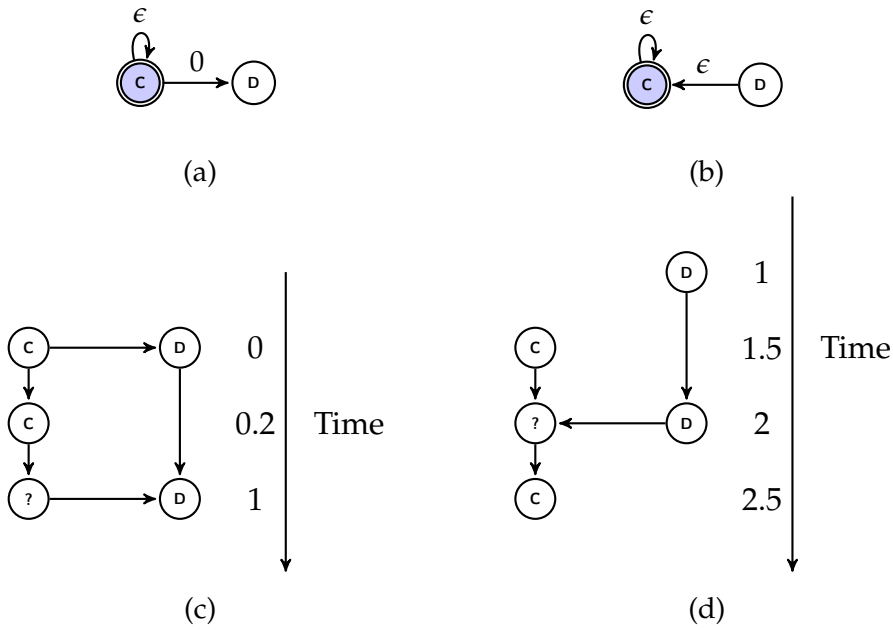


Fig. 4.2: Two possible structures of HTBNs where continuous-time variables are partially observed, whereas discrete-time variables are fully observed: (a) $C \rightarrow D$, (b) $C \leftarrow D$, and their representative BNs given time points $\mathbf{B} = \{0, 0.2, 1\}$ for the former structure and $\mathbf{B} = \{1, 1.5, 2, 2.5\}$ for the latter structure.

Example 4.2

Suppose we have an HTBN with parameters $\boldsymbol{\theta}$, time points $\mathbf{B} = \{0, 0.2, 1\}$, and a directed graph where a discrete component has a parent from a continuous component. At this moment, we discuss the general principles by demonstrating the structure in the simplest situation: an HTBN with a single variable in each component, as illustrated in Fig. 4.2a and its representative BN in Fig. 4.2c. We could also have incomplete data $\boldsymbol{\mathcal{D}}$ over time points $\mathbf{B}$, where the instantiation for variable C at time 1 is missing. Now we can compute the log-likelihood for $\boldsymbol{\mathcal{D}}$ given the HTBN by summing out the log-likelihood for each instantiation c_t and d_t in the data $\boldsymbol{\mathcal{D}}$ where c_t and d_t are the assignments for continuous-time variable C and discrete-time variable D at time t in the data $\boldsymbol{\mathcal{D}}$:

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

Now we turn to another situation where the directionality of the arc between these two components is reversed, i.e., the arc directs from the discrete component to the continuous component, with the resulting graph as shown in Fig. 4.2d. Similarly, we also need to sum out the variable, of which the instantiations is missing in the data $\mathcal{D}$, i.e., the instantiation for variable C at time 2.

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

In order to obtain MAP estimates in this case, we resort to MCMC sampling to maximize the posteriors. We use the probabilistic modeling language Stan, which offers an implementation of MCMC methods as previously described in Section 2.6.2 to iteratively sample parameters from a given distribution. We mainly focus on the description of hybrid time Bayesian networks using the language, in particular describing the log-likelihood of an HTBN given the data on time-points given in Fig. 4.2c. The description of the log-likelihood of an HTBN in Stan

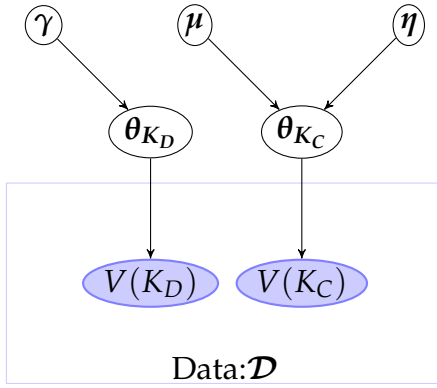


Fig. 4.3: A plate model for an HTBN with continuous-time variables $V(K_C)$ in continuous component K_C and discrete-time variables $V(K_D)$ in discrete component K_D. The prior parameters for variables $V(K_C)$ and $V(K_D)$ are given by γ, and $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$, respectively. Solid filled blue nodes indicate variables, and the rest indicate parameters, including prior parameters.

is depicted by a plate model in Fig. 4.3. In the Stan program, we specified a prior distribution for parameters $\boldsymbol{\theta}_{K_D}$ and $\boldsymbol{\theta}_{K_C}$ with prior γ, and $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ for each discrete component K_D and continuous component K_C, respectively. In addition, a detailed description of log-likelihood of the discrete and continuous component from data $\mathcal{D}$ was given according to Definition 4.14. The MCMC algorithm implemented in the Stan program computed the likelihood for both the discrete and continuous components correspondingly and increases the log-likelihood at each iteration. The MAP estimates for parameters $\boldsymbol{\theta}_{K_C}$ and $\boldsymbol{\theta}_{K_D}$ were derived by the maximum of the distribution of collected samples generated by the algorithm, which is an approximation of the posterior distribution of the underlying parameters.

4.4 EXPERIMENTS

In this section, we experimentally explore learning parameters of HTBNs from complete and incomplete trajectories. First we define an HTBN as a generative model by providing an algorithm that generates the trajectories of the state transitions for variables in the HTBN. Second, we describe the experimental setup, including software packages used in the experiments and the measure used for evaluating learning performance. Third, we present the numerical results for parameter es-

timations of a number of HTBNs from both complete and incomplete data.

4.4.1 *Data Generation Process for HTBNs*

An HTBN can be seen as defining a generative model over sequences of events. In this section, we describe the procedure for generating complete trajectories from a parameterized HTBN, i.e., the states of all variables and time points where a transition occurs are given. Incomplete trajectories are generated by removing a certain amount of values for continuous-time variables at some discrete time points from complete trajectories. The resulting complete and incomplete trajectories are later used for parameter estimation of HTBNs using MAP estimation and Stan sampling. In addition, the trajectories serve as testing data to evaluate the performance of learned models in Section 5.4.2.

The data generation procedure can be seen as generating a sequence of *events*, 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. 4.4 with the following order: Fig. 4.4a $\rightarrow$ Fig. 4.4b $\rightarrow$ Fig. 4.4c $\rightarrow$ Fig. 4.4a, 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. 4.4. 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

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 intensities associated with its current state c^i
      goto 4.4b
    end
  end
  foreach $K_D \in \mathbf{K_D}$ **do**
    goto 4.4c
  end
  $\alpha = s(\alpha)$
end

(a) 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 σ
end
else
  $\beta_{K_C} = s(\alpha)$
end

(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 σ
end

(c) Generate next discrete states

Fig. 4.4: Data generation procedure for HTBNs. 4.4b: sample an event for a continuous component between α and $s(\alpha)$; 4.4c: sample events for discrete components at time $s(\alpha)$.

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. 4.4a and Fig. 4.4b. 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. 4.4c.

4.4.2 *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 $\mathbf{A}$.

We tested on various synthetic data sets, generated from HTBNs according to the procedure defined in Section 4.4.1. 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 test-set generated from the true model. In the following, we fix a large test-set, and evaluate the quality of the model in terms of the distance between the log-likelihood of the data on the original model versus the learned model. Formally:

$$\ell_d = |\ell_t - \ell_e| \tag{4.15}$$

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

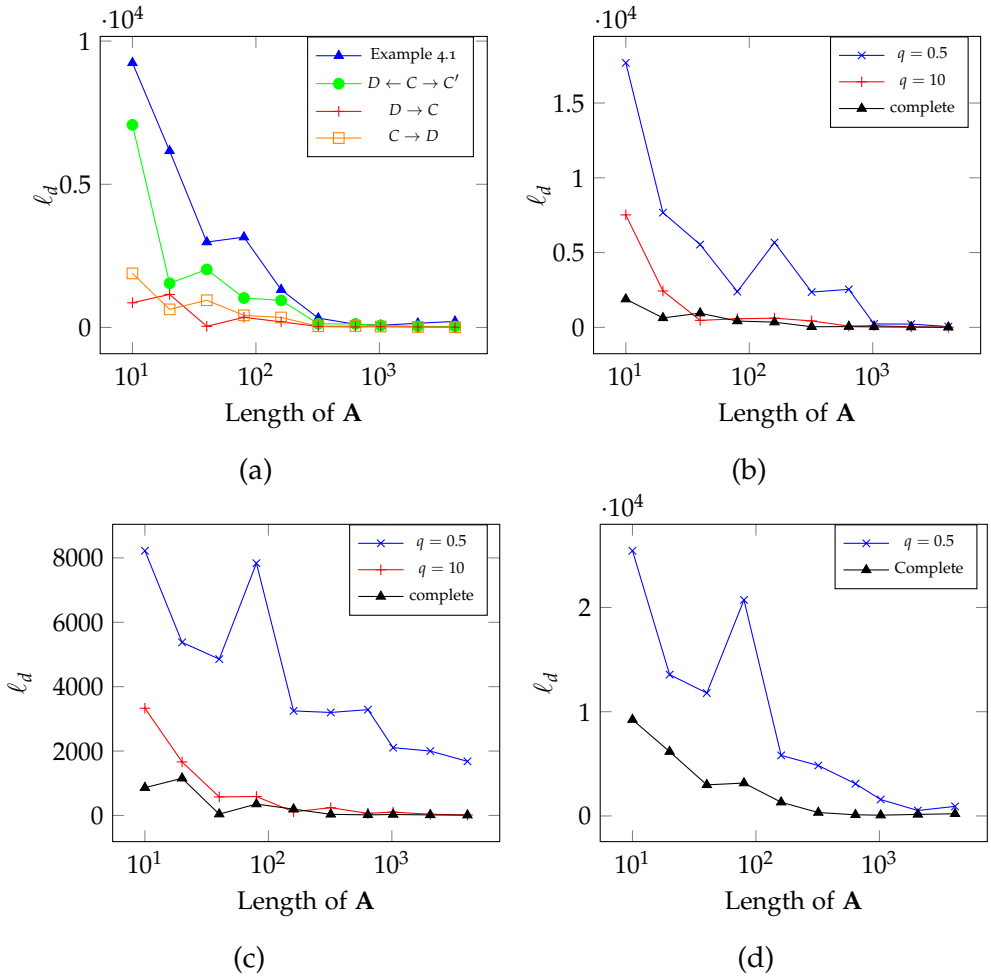


Fig. 4.5: Log-likelihood distance ℓ_d for various HTBNs learned from data. (*a*) Learning with complete trajectories with structure $D \to C$, $C \to D$, $D \leftarrow C \to C'$, and the model in Example 4.1. Figures $(b - d)$ compare learning with complete and partial trajectories for three structures, with (*b*) $C \to D$, (*c*) $D \to C$, (*d*) the HTBN as shown in Example 4.1.

4.4.3 *Results*

We first evaluated the approach to learn from complete trajectories, see Fig. 4.5a. 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. 4.5b and 4.5c, 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 4.1, 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 one can see in Fig. 4.5d, also in this case, models of high quality can be learned using the sampling approach.

4.5 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

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`

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 the 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. Third, an MCMC sampler in CTBNs is recently proposed by Rao *et al.* [76], where the technique of *uniformization* is exploited to approximate a continuous-time Markov process by a discrete-time Markov chain. It is claimed that the sampler gains significant computational benefits over state-of-the-art MCMC samplers for Markov jump process based models, such as CTBNs. It therefore appears worth investigating the computational advantage of using the MCMC sampler in HTBNs, over the one we currently use in Stan.

5

MODELING UNEVENLY SPACED TIME SERIES

Recently, mobile devices, such as smartphones, have been introduced into healthcare research to substitute paper diaries as data collecting tools in the home environment. Such devices support collecting patient data at different time points over a long period, resulting in clinical time-series data with high temporal complexity, such as time irregularities. Analysis of such time series poses new challenges for machine-learning techniques. The goal of the research present in this chapter is to find out which properties of the two types of temporal Bayesian networks, i.e., dynamic Bayesian networks and continuous time Bayesian networks, allow to cope best with unevenly (irregularly) spaced multivariate clinical time-series data.

The capabilities of these temporal Bayesian networks learning from clinical time series that vary in nature are extensively studied. In order to compare the two types of temporal Bayesian network for regularly and irregularly spaced time-series data, three typical ways of observing time-series data were investigated: (1) regularly spaced in time with a fixed rate; (2) irregularly spaced and missing completely at random at discrete time points; (3) irregularly spaced and missing at random at discrete time points. In addition, similar experiments were carried out using real-world chronic obstructive pulmonary disease patient data where observations are unevenly spaced in time.

5.1 INTRODUCTION

The aging of the population is pushing governments and health-care organizations towards improving health-care quality, yet within the boundaries of strict budgetary constraints. At the same time, many governments and health-care organizations are increasingly investing into the use of electronic health technology, often referred as *eHealth*, with the expectation that it will make health-care delivery cheaper, while of-

fering greater control by patients (patient empowerment) [32, 97]. The general trend in most parts of the world is that health-care costs are increasing, sometimes quite steeply, and eHealth is seen as a way to move part of the health care burden from expensive institutional organizations, such as hospitals, to the home environment, thus contributing to the reduction in health-care costs.

It is now becoming clear that eHealth is shifting the health-care field to an increasingly data-driven way of working, yielding substantial *quantities* of patient data. However, the data-driven paradigm also renders the *quality* of the collected data of paramount importance to build and deploy sufficiently accurate models that support both patients and doctors. A common means to collect data in many clinical studies are paper diary cards [8, 44]. Patients are encouraged to fill out diary cards, thereby documenting the status of their health-related symptoms in the form of their responses to a questionnaire. Major drawbacks associated with using paper diary cards, in general, are that the dates and times of paper diary entries are often missing, due to the patient's poor compliance [90]. Therefore, the quality of the data collected from paper diaries has its limitations.

Another drawback of using paper diaries is the lack of generalizability. The time points of observations collected by paper diaries can be viewed as regular random samples from the timeline with a certain rate, also known as *observation rate*, e.g., once every day. However, having a fixed observation rate restricts eHealth studies to shorter periods and a smaller scale. It is unrealistic to expect that many patients are willing to collect clinical data on a regular basis as part of a long-term study. A more realistic assumption is that the regularity of recording an observation by the patient will vary and may be affected by many factors, such as whether or not the patient feels ill. This implies that methods to handle different time-regularity patterns are greatly needed.

Besides differences in time regularity, time *irregularity* is another common phenomenon of clinical time series. In clinical trials, the patient's health status, in terms of physiological data, may be observed only at irregularly spaced points in time. In addition, it is very unlikely that different patients are observed at the same points in time. Most of the current literature is based on statistical analysis of periodic snapshots of physiological measurements with a fixed time interval, such as daily [41, 42] or weekly [8]. In this chapter, we aim to learn accurate and use-

ful models from irregularly spaced clinical time series using temporal Bayesian networks.

To provide a more concrete clinical context for this research, we pay attention to chronic obstructive pulmonary disease (COPD) as an application area. COPD is a progressive disease where a patient's deterioration manifests itself in worsening symptoms, known as an *exacerbation*. It is of clinical interest to predict whether and when an *exacerbation event* will occur for a given patient. However, an exacerbation can not be directly observed. It is defined either in terms of specific worsening symptoms for consecutive days or if there is evidence of a patient's hospital admission due to an exacerbation. Unfortunately, clinicians have so far not been able to agree on a clinical definition of an exacerbation [8, 20].

Rather than focusing our research on automatically deciding on the presence or absence of an exacerbation, using multiple definitions, we aim at trying to understand the dynamic behavior of the symptoms of COPD. The advantage is that we do not have to bother about the lack of a definition of a COPD exacerbation.

The main contribution of our work consists of two parts. One contribution lies in capturing COPD symptom dynamics, which we see as representative for many other diseases that are being monitored in the home environment. So far it is unknown which particular method best captures disease dynamics using data from home monitoring. The second contribution lies in the in-depth investigation of two temporal Bayesian network methods to model the dynamics: *dynamic Bayesian networks* (DBNs), where time is assumed to be discrete, and *continuous-time Bayesian networks* (CTBNs), where time is assumed to be continuous. We also believe that this study sheds some light on the practical requirements of using DBNs and CTBNs in general.

The performance of DBNs and CTBNs for modeling the dynamics of COPD symptoms is investigated given COPD time series in three forms:

- when observations are made regularly at time points but with different observation rates;
- when time points of observations are unevenly spaced over time as a consequence of two missing data mechanisms, i.e.,
  (1) the probability of having variables observed at a time point is independent from other time points where variables are

observed or unobserved, also known as *missing completely at random* (MCAR);

(2) the probability of having variables observed at a time point is dependent on other time points where variables are observed, also known as *missing at random* (MAR). More specifically, the values for variables in the system are missing at time $t+1$ if the values at time $t+1$ are identical to those at time t.

In the rest of the chapter, we only focus on the situation where variables are either fully observed or completely missing at a given time point. We investigate the performance of DBNs and CTBNs to learn from regular and irregular COPD time series. Within CTBNs, we also study the impact of the *evidence type*, i.e., point and interval evidence, and hyper-parameters on the performance of CTBNs.

To the best of our knowledge, this is also the first work where hyper-parameters in CTBNs are taken into consideration in the modeling process. Within DBNs, we study the performance of DBNs interpreting time series in three ways, namely, (1) viewing time series as a sequence; (2) imputing values at discrete time points with the *Last-Observation-Carried-Forward* (LOCF) method (See Section 5.3.4); (3) filling in missing values at discrete time points by *Expectation Maximization* (EM). Our final aim is to gather information about potential factors that practitioners of temporal Bayesian networks need to take into account to learn a model from unevenly spaced clinical multivariate time series.

The rest of the chapter is organized as follows. In the following section, we devote ourselves to describing the related work about predicting COPD exacerbation using machine-learning techniques and the state of the art of continuous time Bayesian networks. It is followed by a detailed description of a synthetic and real-world time series used to conduct experiments, and a number of ways to interpret irregularly spaced observations and the choice of hyper-parameters in CTBNs in Section 5.3. The experimental setup is described in Section 5.4.1, including software packages used in the experiments and evaluation methods. Comprehensive results are given in Section 5.4.2, where we compare the performance of dynamic Bayesian networks and continuous time Bayesian networks both for simulated time series and for a real-world time series. Finally, we discuss our work's contribution, limitation, and future work in Section 5.5.

5.2 RELATED WORK

5.2.1 *The Clinical Setting: COPD Symptomatology*

The availability of a widely accepted definition of an exacerbation of COPD in the medical community would definitely help to facilitate public communication and designing guidelines. Unfortunately, as said above, such a definition is still not available [20, 84, 101]. There is some work in the literature that studies the diagnostic impact of various definitions of an exacerbation [8, 42]. So far, clinicians use a variety of clinical features to describe the COPD-related health status of a patient [44, 94]. Even for the most accepted definition of an exacerbation at this moment, the *Anthonisen criteria* (AC) [4], the required major and minor symptoms are not always available partly due to design of the clinical study (see [42]). In [46, 83], an exacerbation is defined in terms of a patient's hospital admission, or a non-scheduled visit to the emergency unit or to the specialists because of respiratory symptoms, or self-treatment of the patient by antibiotics. Because of their limited medical knowledge, patients are prone to misuse of antibiotics, i.e., they use antibiotics for a viral infection or a bacterial non-pulmonary infection. Instead, work in [41] chooses the worsening of respiratory symptoms at two consecutive days as an indicator of an exacerbation. It is clearly easier to predict an exacerbation for the next day when an exacerbation is currently observed than when it has not been observed. However, it seems that the authors provide no clear distinction between these two situations.

In the context of COPD management, a telehealth system [6] has been described previously that supports decision making. However, its decision support is limited to rule-based detection of abnormal values and to simple trend analysis. In contrast, predicting a COPD exacerbation, i.e., when the patient's health condition gets worse, can help to support the patient and doctor by providing an opportunity for early intervention before it is too late. To this end, some work has focused on the development of classifiers, e.g., using K-nearest neighbors (K-NN) [46] and K-means clustering [83] to predict the onset of an exacerbation given the patient's signs and symptoms. Nevertheless, there is still a lack of an explicit description of the underlying dynamics of the clinical symptoms in these models. Capturing temporal dynamics of signs and symptoms is the main goal of [41], where time and uncertainty are also considered

for the first time. Given a limited amount of temporal clinical data, the work chooses to use dynamic Bayesian networks to capture the dynamics of COPD symptoms. As a consequence, the approach suffers from the need of finding the finest time interval. Usually this is undesirable both from a modeling and inference perspective. For example, the models described above are unable to capture the dynamics of symptoms [46, 83], or they do not take time as a parameter [41].

We conclude that a data-driven temporal model capturing the COPD dynamics is of clinical interest. It would not suffer from the subjective nature of a definition of an exacerbation, and may yield much more valuable insight into the nature of COPD in comparison to what can be achieved by a classifier model.

5.2.2 *Model Development: Temporal Bayesian Networks*

In the previous section, we have already mentioned the related work by Van der Heijden et al. [41], which uses dynamic Bayesian networks (DBNs) for the detection of exacerbations of COPD. Another way to model the dynamics of symptoms using Bayesian networks is offered by continuous time Bayesian networks, i.e., time is used as a continuous parameter. The states of the symptoms satisfy a multinomial distribution, whereas the time when a transition occurs, e.g., a symptom changes from one state to another, is modeled as an exponentially distributed parameter. Early work has demonstrated the powerful expressiveness of CTBNs to model the dynamics of systems where variables are observed at time points that are unevenly spaced over time [10, 102, 103]. In the specific domain of medical applications, CTBNs have been used to diagnose cardiogenic heart failure and have been shown to anticipate its likely evolution [26, 55]. They have also been used to construct gene networks [1] to generate hypotheses for biological experiments [2]. Nevertheless, the current clinical applications significantly suffer from the unavailability of temporal patient data. The quantitative component of the CTBN model in [26], i.e. the parameters, are so far mainly elicited on the basis of clinical expertise. In our work, however, CTBNs are both applied to clinical synthetic data and real-world data.

Like standard Bayesian networks, evidence in CTBNs is also associated with a probability to incorporate the uncertain nature of observations [92]. In addition, evidence entails the amount of time that a variable stays in a state. While *point evidence* claims that a variable holds a

value for an infinitely small amount of time $\Delta t \to 0$, *interval evidence* states that a variable in the system remains in a certain state throughout an interval of time. The concept of interval evidence is firstly introduced in [29], where it is originally called negative evidence.

Another relevant extension of Bayesian networks are irregular-time Bayesian networks [75]. These models aim to increase the expressiveness of the temporal dynamics to handle irregular time series, with variables having a continuous state space. In the present work, however, we focus on discrete state spaces, which are typical for CTBNs and DBNs. Acerbi *et al.*[1] attempt to study the difference in performance between DBNs and CTBNs in a specific problem in the realm of molecular biology, where gene expressions are unevenly distributed over time. In their work, the focus is on the reconstruction of a gene network using DBNs and CTBNs, where solely simulated gene data are used. In addition, it still remains unclear whether there is a difference in the performance of CTBNs when using point and interval evidence. Our work, however, clarifies this difference in the practical use of CTBNs.

5.2.3 *Data: Irregular Longitudinal Clinical Data*

The temporal representation of clinical data has been extensively investigated by researchers in Artificial Intelligence in Medicine for more than two decades [3, 11, 47, 48]. Most of this research deals with the use of time in clinical reasoning, e.g., for treatment planning and decision support, which is not of immediate relevance for our research. However, the reason why there is so much research on temporal reasoning in medicine is due to the significance of time in medical decision-making. In the context of the current research, we are dealing with a special kind of clinical temporal data, i.e., data that are being recorded by patients at home.

In many longitudinal clinical trials, patients are followed over a period of time and are scheduled to be assessed at a prespecified visit time after being enrolled in the study. However, patients often selectively miss their visits or return at non-scheduled points in time. As a result, the times of measurements are irregular, yielding a highly imbalanced time series. Some medical examples of this phenomenon are given by studies on the incident rate of sexual maturation [21] and by homeless people with mental illness [52].

Algorithm 5.1: Generating regular time series D_{REG}.

Data: A time series $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the dataset D_L, where n is the number of observations in D_L; observation rates $R = \{1, 2, 3, 4, 5, 6, 7, 14, 21, 28\}$;

Result: D_{REG}

$D_{\text{REG}} = \varnothing$;
foreach $r \in R$ **do**
  $D_r = \varnothing$;
  foreach $i \in \{1, \ldots, r\}$ **do**
    Create a time series S_i with observations $\mathbf{x}_i, \mathbf{x}_{i+r}, \ldots, \mathbf{x}_{i+mr}$, where $m = \max\{s \mid i + sr \leq n\}$;
    $D_r = D_r \cup \{S_i\}$;
  end
  $D_{\text{REG}} = D_{\text{REG}} \cup \{D_r\}$;
end
return D_{REG};

Advances in computer technology have turned mobile devices into efficient data collecting tools in many scientific disciplines. One advantage of a mobile, network-linked digital tool is that patients are less likely to miss the time window of taking a measurement. However, the novel tools also create new mechanisms for obtaining irregular time series. One characteristic of the new mechanism is that all measured variables are either fully observed or fully missing at a give time point. The apparent irregularity pattern can be, in part, due to patients being instructed to report all their symptoms only when abnormal symptoms are detected.

5.3 MATERIALS

In this section, the time-series datasets used in the research are described.

5.3.1 *Synthetic Datasets*

Data of a COPD patient cohort in London, which we obtained from the research group of Wedzicha *et al.* [44], were employed to generate time

	Time (day(s))									
	1	2	3	4	5	6	7	8	9	10
D	0	0	0	0	1	1	1	0	0	0
SV	0	0	0	0	1	1	1	0	0	0
SC	0	0	0	0	1	1	1	0	0	0
W	0	0	0	0	1	1	1	0	0	0
C	0	0	0	0	1	1	1	0	0	0
Temp	0	0	0	0	0	0	0	0	0	0
O	0	0	1	0	1	1	1	0	0	0

(a)

	Time (day(s))				
	1	3	5	7	9
D	0	0	1	1	0
SV	0	0	1	1	0
SC	0	0	1	1	0
W	0	0	1	1	0
C	0	0	1	1	0
Temp	0	0	0	0	0
O	0	1	1	1	0

(b)

	Time (day(s))				
	2	4	6	8	10
D	0	0	1	0	0
SV	0	0	1	0	0
SC	0	0	1	0	0
W	0	0	1	0	0
C	0	0	1	0	0
Temp	0	0	0	0	0
O	0	0	1	0	0

(c)

Fig. 5.1: A fragment of dataset D_L over ten consecutive days (a); a dataset D_{REG} consists of time series S_1 (b) and S_2 (c) generated from the fragment given in (a) according to the Algorithm 5.1 where $r = 2$. For an explanation of the meaning of the mentioned variables: see text. The value '0' stands for normal and '1' for abnormal.

series with variables observed at equally and unequally spaced time points. The symptoms and signs in the original dataset were recorded by the patients on a daily basis. The methodology of the data collection process was previously extensively discussed in [86]. An earlier attempt to capture the temporal interactions in these data was made by van der Heijden *et al.* [41]. The dataset, denoted as D_L (where 'L' stands for 'London'), consists of time series of thirteen COPD patients; each of them had at least one exacerbation. The data contains a total of 2,849 data entries with values for the variables dyspnea (D), sputum volume (SV) and purulence (SC), wheeze (W), cough (C), temperature ($Temp$), and oxygen saturation (O). A fragment of dataset D_L over ten consecutive days is shown in Fig. 5.1a.

To study the behavior of CTBNs and DBNs for time series $\{\mathbf{x}_t \mid t \in T\}$, where $\mathbf{x}_t$ denotes a vector of values for the variables (symptoms and signs of COPD) at time point t, we generated time series from the dataset D_L given an *observation rate* r: the time interval between two successive observations. Ten time series were generated according to the algorithm described in Algorithm 5.1, collectively denoted as D_{REG}, with the observation rate ranging from 1–6 day(s) to 1–4 week(s). For illustrative purpose, a dataset D_{REG} is generated from the fragment given in Fig. 5.1a by Algorithm 5.1 where $r = 2$, consisting time series S_1 and S_2 given in Fig. 5.1b and Fig. 5.1c

Algorithm 5.2: Generating irregular time series D_{MCAR}.

Data: A time series $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the dataset D_L, where n is the number of observations in D_L; % of removed entries $P = \{5, 10, 30, 50, 60, 70, 80, 90, 95\}$;

Result: D_{MCAR}

1 $D_{\text{MCAR}} = \varnothing$;
2 **foreach** $p \in P$ **do**
3 Randomly sample a set of time index D from $\{1, 2, \ldots, n\}$, where $|D| = n * p\%$;
4 $S = \{1, 2, \ldots, n\} \setminus D$;
5 Select observations from D_L with time indices S, yielding time series D_S;
6 $D_{\text{MCAR}} = D_{\text{MCAR}} \cup \{D_S\}$;
7 **end**
8 **return** D_{MCAR};

Algorithm 5.3: Generating irregular time series D_{MAR}.

Data: A time series $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the dataset D_L, where n is the number of observations in D_L;
Result: D_{MAR}

```
D_MAR = {x_1};
foreach i ∈ {2, ..., n} do
    if x_i ≠ x_{i−1} then
        D_MAR = D_MAR ∪ {x_i};
    end
end
return D_MAR;
```

Similarly, we investigate the capability of CTBNs and DBNs to handle time *ir*regularities by generating irregular time series. First, we generated time series $\{\mathbf{x}_t \mid t \in T\}$, where observations are made at discrete-time points that are completely randomly and irregularly sampled in time. Nine datasets, collectively denoted as D_{MCAR}, were generated by randomly removing entries with percentage of 5, 10, 30, 50, 60, 70, 80, 90, and 95 from the dataset D_L, as shown in Algorithm 5.2. An illustrative dataset D_{MCAR} as shown in Fig. 5.3a is generated from the fragment given in Fig. 5.1a with $p = 50$. Second, we generated one time series where the missingness is dependent on the observations. We removed consecutive identical entries in the dataset D_L (see Algorithm 5.3), resulting in an irregular time series D_{MAR}. The number of observations for thirteen patients in dataset D_{MAR} is given in Fig. 5.2. The corresponding D_{MAR} for the fragment described earlier in this section is given in Fig. 5.3b.

Patient ID												
1	2	3	4	5	6	7	8	9	10	11	12	13
111	209	173	145	257	189	145	155	147	203	152	198	137

Fig. 5.2: The number of observations in the dataset D_L for thirteen patients where there are significantly more observations indicated by the red text for patient indexed by 5 than the other twelve .

	Time (day(s))				
	1	3	5	6	8
D	0	0	1	1	0
SV	0	0	1	1	0
SC	0	0	1	1	0
W	0	0	1	1	0
C	0	0	1	1	0
Temp	0	0	0	0	0
O	0	1	1	1	0

(a)

	Time (day(s))				
	1	3	4	5	8
D	0	0	0	1	0
SV	0	0	0	1	0
SC	0	0	0	1	0
W	0	0	0	1	0
C	0	0	0	1	0
Temp	0	0	0	0	0
O	0	1	0	1	0

(b)

	Time (day(s))							
	1	3	4	5	6	7	9	10
D	0	0	0	1	1	1	0	0
SV	0	0	0	1	1	1	0	0
SC	0	0	0	1	1	1	0	0
W	0	0	0	1	1	1	0	0
C	0	0	0	1	1	1	0	0
Temp	0	0	0	0	0	0	0	0
O	0	0	1	0	1	1	0	0

(c)

Fig. 5.3: Illustrative examples of generated datasets from the fragment of D_L given in Fig. 5.1a according to Algorithm 5.2, Algorithm 5.3 and Algorithm 5.4 respectively: (a) D_{MCAR} where $p = 50$ and selected time indices $S = \{3, 4, 6, 7, 10\}$; (b) D_{MAR}; (c) D_{L2A} with duplicated entries at time 2 and 8 removed.

5.3.2 *ACCESS Dataset*

A real clinical dataset, independent of the London dataset discussed in the previous section, was subsequently used to study the behavior of DBNs and CTBNs on real-world irregular time-series. The dataset was collected using the recently developed *"Adaptive Computerized COPD Exacerbation Self-management Support" (ACCESS)* [1] system by Radboud University. The data were collected over two years for 40 patients, aged between 46 and 89 years (mean (sd): 69.5 (8.9)), from June 2015 to February 2017. All patients were recruited from hospitals and primary care practices in the Netherlands based on their willingness to participate in a long-term study. Inclusion criteria were: post-bronchodilator $FEV_1/FVC < 0.70$; at least 2 self-reported exacerbations in the 12 months preceding the time of recruitment and no severe co-morbidity. Diseases such as diabetes, kidney diseases, and smoking habits were also recorded.

The ACCESS dataset, denoted as D_A, consists of a total of 1,138 data entries with the same variables as in D_L. As part of the training, the patients were instructed to daily self-report using the ACCESS system for the first two weeks. As a consequence, the observations at the beginning of the study are relatively regularly spaced over time with a time interval of a day. Later in the study, most patients were reluctant to comply to the registration of their health status on a daily basis over a period of many months. To deal with this problem, after the two-weeks initial registration, the patients were instructed to take the initiative to make a registration of symptoms and signs when they detected something abnormal in their symptomatology. However, not all patients followed exactly the instruction, in particular, three patients randomly registered their respiratory symptoms in the entire study period. As a consequence, the time intervals between two consecutive observations (mean (sd): 7.4 (21.5) days) varied considerably from patient to patient.

Given an irregular time series for a specific medical problem, adopting an appropriate temporal technique requires a better understanding of the cause of the time irregularity. For the irregular time series D_A from the ACCESS study, it is reasonable to assume that the patients did not encounter any worsening symptoms for the days when no values were filled out for the variables. This is in accordance with the instructions they received at the beginning of the study which stated that symptoms only had to be recorded when something abnormal occurred.

1 see https://clinicaltrials.gov/ct2/show/NCT02553096

Algorithm 5.4: Generating irregular time series D_{L2A}.

Data: A time series $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the dataset D_L, where n is the number of observations in D_L and time series D_A;
Result: D_{L2A}

```
Calculate the percentage of missing entries in D_A, denoted as p;
S = ∅;
foreach i ∈ {1, ..., n} do
    if x_i indicates the symptoms are normal then
        S = S ∪ {x_i};
    end
end
Random select a subset D_p from S with the number of entries p * n;
D_L2A = D_L \ D_p;
return D_L2A;
```

To have a better understanding of the behavior of CTBNs and DBNs to handle such time irregularity, we generated another irregular time series in accordance to time series D_A. More specifically, we removed the same amount of entries where symptoms were normal from D_L in comparison to time series D_A, resulting in a time series denoted as D_{L2A} ('London to ACCESS', see Algorithm 5.4). An illustrative D_{L2A} shown in Fig. 5.3c is generated from the fragment given in Fig. 5.1a where the duplicated entries at time 2 and 8 are removed.

Note that variables in all synthetic and real-time series, namely, D_L, D_{REG}, D_{MCAR}, D_{MAR}, D_{L2A}, and D_A, are either fully observed or fully missing at a given time point.

5.3.3 *Interpretation of Unevenly-spaced Time Series*

A *time series* is a set of observations $\{X_t = x \mid t \in T\}$, where the observation of variable X takes the form $X_t = x$, with x the value at time t. Unlike sequential data, the actual time stamp t is an important aspect of a time series. For example, we may have a time series recording whether or not a patient coughs at discrete time points 9:00, 11:00, 11:30, 12:00 in the morning.

In the statistical literature, there is some recent work on converting unevenly-spaced time series to equally-spaced data, or to directly analyze and manipulate the unevenly-spaced time series without an

equally-spaced transformation (see e.g. [19, 77]). In this chapter, we will consider some natural choices when employing probabilistic graphical models for analyzing unevenly spaced clinical time series, in particular by transformation (e.g. by imputing the last observation) or by directly analyzing the irregular time series.

5.3.4 *Interpretation of Time Series by DBNs*

For discrete-time models, we consider three ways for interpreting a time series. An unevenly-spaced time series can be directly viewed as just a sequence, ignoring time differences between consecutive observations. Alternatively, the occurrence of a transition can be specified at discrete time points with a fixed time interval. In this case, two common methods are considered to handle irregular time series, as there will be missing data at some of the discrete time-points. These methods are (1) Expectation Maximization (EM) to learn from time series with missing data; (2) imputing the last observation (Last Observation Carried Forward, LOCF): values are filled in for variables at discrete time points where there is missing data.

Example 5.1

Consider a time series of observations for coughing at time points 9:00, 11:00, 11:30 and 12:00. The time series can be interpreted as a sequence by discarding the time stamps as shown in Fig. 5.4a, i.e., we assume we have complete data for this sequence of observations. If we select a half hour as the fixed time interval, the time series then contains a number of missing data points as shown in Fig. 5.4b. Using this data, EM can be applied to learn from this time series directly. Imputing the last observation in the time series, which again leads to complete data (see Fig. 5.4c).

5.3.5 *Interpretation of Time Series by CTBNs*

In continuous-time models, there are two natural choices for interpreting time series. An observation is interpreted as *point evidence* when it only describes momentary behavior of a variable or system. For example, a patient is observed to have a cough at a certain point of time during the day. The counterpart of point evidence, *interval evidence*, on

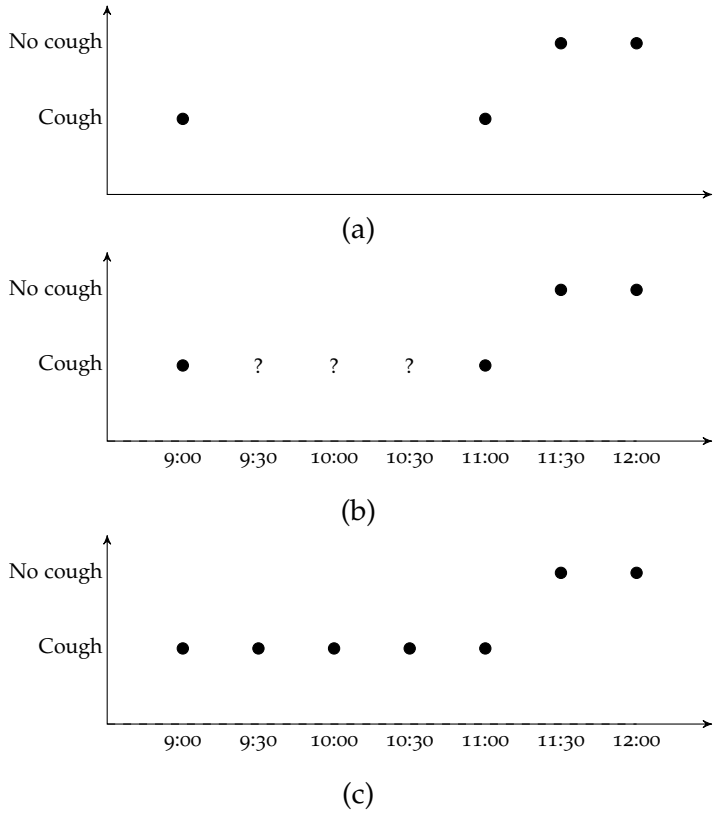


Fig. 5.4: Three DBN interpretations of clinical time series for the COPD symptom cough at time points 9:00, 11:00, 11:30 and 12:00. (a): just as a sequence with time stamps discarded; (b) missing data at some of the discrete time points are indicated by a question mark '?'; (c) with imputing the last observation at some of the discrete time points with missing data.

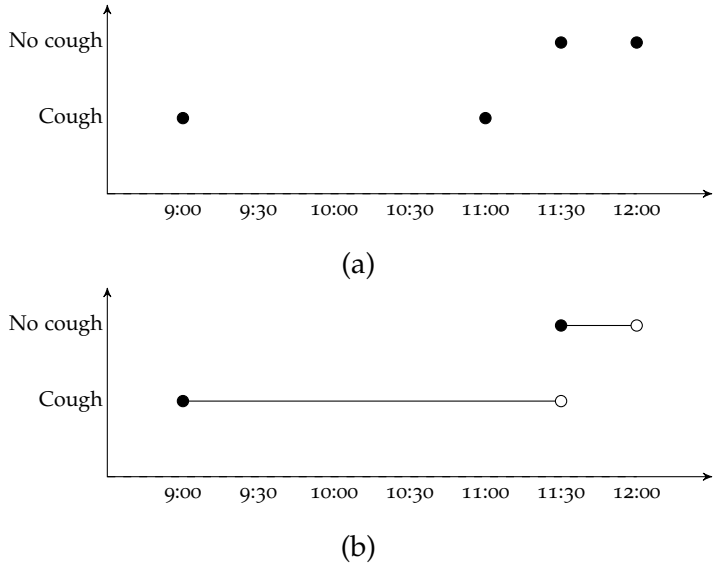


Fig. 5.5: Two CTBN interpretations of clinical time series for cough at time points 9:00, 11:00, 11:30 and 12:00. The observations are interpreted as point evidence in (a) and as interval evidence in (b).

the other hand, states that a variable stays in a state for a given period of time. Formally, interval evidence on a right-open interval $[t_1, t_2), t_1 < t_2$, can be denoted by $X_{[t_1,t_2)} = x$, asserting that X stays in state x during the given time interval $[t_1, t_2)$.

Example 5.2

Reconsider the time series from Example 5.1. The observations can be interpreted as point evidence (see Fig. 5.5a) or as interval evidence (see Fig. 5.5b). In the former, the interpretation states that the patient only coughs at time point 9:00, 11:00, 11:30 and 12:00. In the latter, however, the interpretation states that the patient keeps coughing in the time intervals $[9{:}00, 9{:}30)$ and $[11{:}00, 11{:}30)$ and does not stop in the time interval $[11{:}30, 12{:}00)$.

5.3.6 *Choice of Hyperparameters in CTBNs*

An issue that arises when learning CTBNs from clinical time series is choosing values for the associated hyperparameters. CTBN behavior is characterized by hyperparameters that are distinct from those of other

temporal probabilistic models; together they give a prior estimate of the time that a variable stays in a given state. When a time series is sufficiently large to capture all possible transitions, in particular for variables which change at a very slow rate, the hyperparameter τ has little impact on the learned models. However, this appears to be a rare situation for clinical time series. For chronic diseases, such as COPD in our case, a change of relevant symptoms can take so much time that it will never be observed in the limited clinical datasets that normally are available. This shortcoming of clinical time series is, in part, due to the difficulty of collecting data for many patients with *sufficient temporal detail* during a long period of time.

While some search algorithms have been devised to optimize hyperparameters (see e.g. [60, 96]), they often come at the expense of high computational costs. Instead, a more cost-efficient approach can be used to constrain the state space of hyperparameters by the utilization of domain knowledge. In the present research, we use such an approach to select an appropriate hyperparameter for CTBNs to model the dynamics of COPD symptoms. Theoretically speaking, one symptom might have a hyperparameter configuration that differs from that of the other ones. In the chapter, however, it is assumed that all COPD symptoms have the same configuration.

For COPD, we hypothesized that it is reasonable to assume that a change of symptoms expressed by the hyperparameter τ (See Section 2.5.2) takes less than 100 days but no less than 0.1 day. Therefore, we considered six possible values, i.e., $\tau \in \{0.1, 1, 10, 20, 50, 100\}$. Irrespective of the difference in value of the hyperparameter τ, the other hyperparameter α is set to 1. In Fig. 5.6, we present the result of this experiment where we subtract the log-likelihood from CTBNs with the hyperparameter having the value of 10 by those having one of the other five values. The results also show the impact of time granularity on the subtracted log-likehood. The results suggest that there is a relatively smaller difference in terms of log-likelihood for CTBNs using the values between 1 and 20. The results indicate that CTBNs in general achieve the best performance using the value of 20 for the hyperparameter τ in the given five choices. Neverthess, they are still outperformed by those learned by using $\tau = 10$. Thus, in the following experiments in which CTBNs are compared to DBNs, the hyperparameter τ is fixed to 10, both when learning CTBNs from regular and irregular clinical time series.

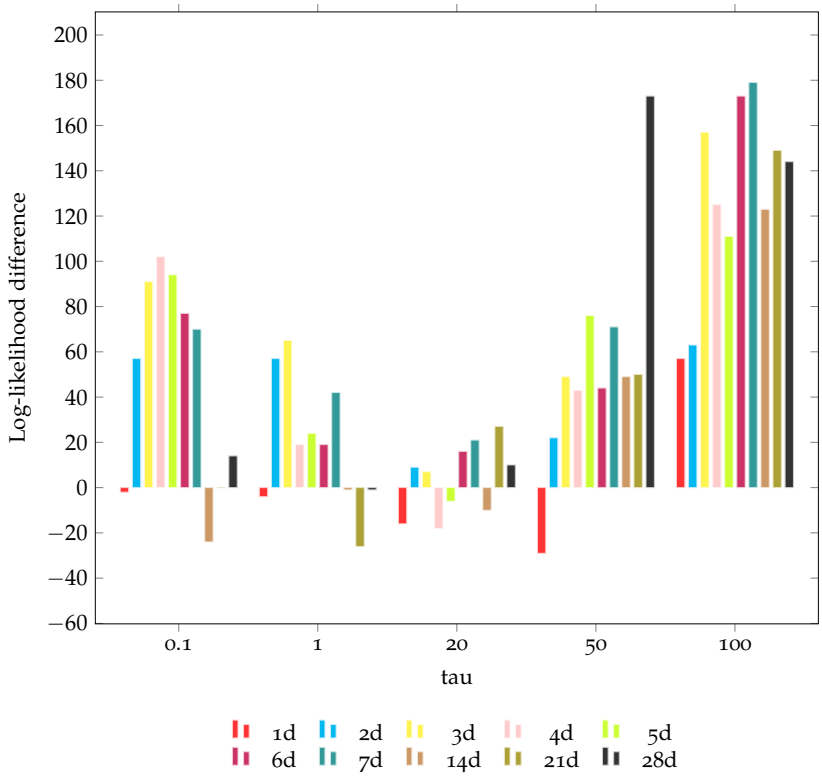


Fig. 5.6: Log-likelihood difference of CTBNs with interval evidence between a value of $\tau = 10$ and other values of $\tau \in \{0.1, 1, 20, 50, 100\}$ for various time granularities (1–7 days and 2–4 weeks). A higher positive value in log-likelihood indicates a performance for a CTBN with a value for $\tau \neq 10$ that is worse in comparison that that for $\tau = 10$.

5.4 EXPERIMENTS

In this section we describe the experimental setup for learning and evaluating temporal probabilistic models, CTBNs and DBNs, of the evolution of COPD symptoms and signs.

5.4.1 *Experimental Settings*

The purpose of the experiments is, firstly, to obtain insight into the behavior of CTBNs and DBNs for synthetic data, where observations are made at time points that are (1) equally spaced, and (2) unevenly spaced; secondly, the model types were also studied for their capability to handle time irregularity in a real-world situation. For this purpose, we generated several synthetic datasets, as described in Section 5.3.1, from an existing dataset that contained daily data, and consider one real-world time-series dataset, described in Section 5.3.2, for which patients entered data in an irregular way.

A number of software packages and tools were used in the experiments. With respect to learning DBNs, tools needed to learn from regular and irregular clinical time series were different. For the former, we first used *Banjo*[2] to learn a structure from regular time series D_{REG}, and subsequently used *bnlearn*[3] package in R to learn its parameters. For the latter, a choice to learn both structure and parameters using EM may seem reasonable. However, such an approach has been shown to offer limited capability to recover the underlying dependences between random variables [41, 104]. In this paper, we chose to predefine a unique structure for the remaining irregular time series. We chose to learn the structure from the regular time series D_L, which ensures that differences in performance between models are not due to a poor structure obtained from using the EM algorithm. Given the learned structure, we used BNT tools[4] with the implementation of EM to learn parameters from irregular time series, i.e., D_{MCAR}, D_{MAR}, D_A, D_{L2A}. Furthermore, the R interface[5] for *"Continuous Time Bayesian Network Reasoning and Learning Engine (CTBN-RLE)"* was used to learn both structure and parameters for CTBNs.

2 https://users.cs.duke.edu/∼amink/software/banjo
3 http://www.bnlearn.com
4 https://www.cs.utah.edu/∼tch/notes/matlab/bnt/docs/bnt_pre_sf.html
5 http://rlair.cs.ucr.edu/ctbnrle/Rinterface

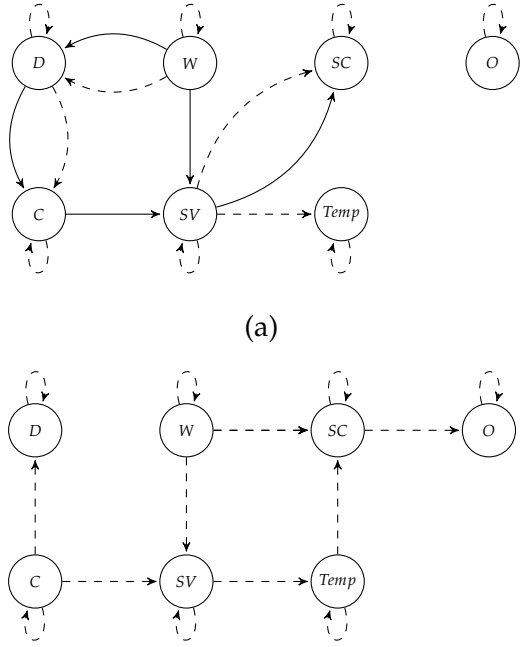


Fig. 5.7: Learned structures of a DBN and a CTBN from the London dataset with seven variables: O) Oxygen saturation; SC) sputum purulence; W) Wheeze; D) Dyspnea; C) Cough; SV) sputum volume; Temp) temperature. (a): A DBN; (b): A CTBN, where the value of hyperparameter τ is set to 20. Solid arcs indicate atemporal dependence and dashed arcs temporal dependence.

To prevent overfitting, a K-fold cross-validation procedure was used where the data was randomly split into K partitions, with $K-1$ partitions for learning and one partition for testing. The data for testing did not contribute to the learning of models. The number of folds K was set to 13 (the number of patients in the London data cohort) for time-series D_{REG}, D_{MCAR}, D_{MAR}, D_{L2A} and to 10 for the time series D_A, respectively. For each fold, the performance of DBNs and CTBNs was evaluated in terms of log-likelihood. However, the evaluation of CTBNs with point evidence was slightly different as the CTBN learning algorithm using point evidence led to a significant variation in the quality of the learned models. For this reason, we used an additional validation set to select a good-quality CTBN model when using point evidence.

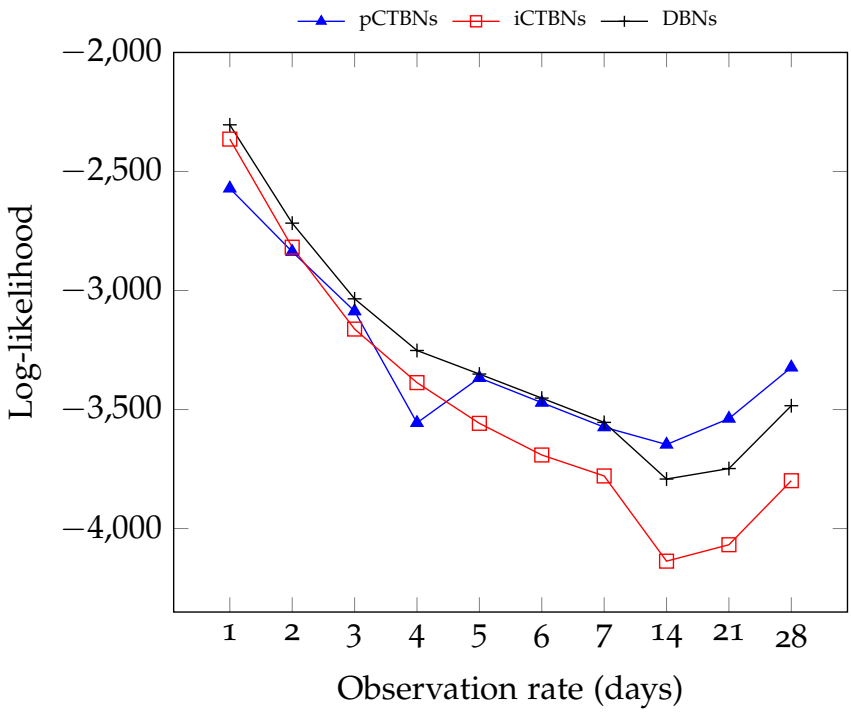
5.4.2 *Results*

In this subsection, we will investigate and discuss the performance of DBNs and CTBNs when learned from regular and irregular time series. More specifically, their learning performance in terms of log-likelihood will be studied using a number of both synthetic and real-world time-series datasets. In the following, we will use the notation *iCTBN* and *pCTBN* for a learned CTBNs using interval and point evidence, respectively, and *sDBN*, *iDBN* and *emDBN* for DBNs learned from a sequence, from an imputed dataset, and using the EM algorithm, respectively.

5.4.2.1 *Results for Synthetic Data*

Synthetic time series are valuable for obtaining an understanding of how well CTBNs or DBNs can capture temporal knowledge when learning from a regular or irregular time series, as it is clearly impossible to obtain real-world time series that conform to any possible temporal pattern. In that sense, learning DBNs and CTBNs from synthetic data can act as a benchmark. The results were obtained by using the clinical time series previously described in Section 5.3.1.

REGULAR DATA WITH DIFFERENT OBSERVATION RATES Given regular time series, we first study the impact of variations in observation rate on the performance of both DBNs and CTBNs. We learned *both the structures and parameters* from the data; the learned DBN and CTBN network structures, with the observation rate set to one day, are shown in Fig. 5.7. The results for the various models are shown in Fig. 5.8, where the log-likihood is the sum of those for all the thirteen folds. A decrease in the number of observations from the time series D_{REG} accounts for an increase in the log-likelihood if the observation rate is higher than 14 days. Overall, it is not surprising that the results show a declining performance for both DBNs and CTBNs when the observation rate increases, as it makes it more difficult to learn the underlying transition probabilities. Irrespective of the observation rate, the results also show that DBNs have a higher performance than iCTBNs in terms of log-likelihood, although the differences do not reach statistical significance (see Fig. 5.8b). When the observation rate is not larger than a week, DBNs also have a higher log-likelihood than pCTBNs. Otherwise, pCTBNs have a better performance. In both cases, the difference



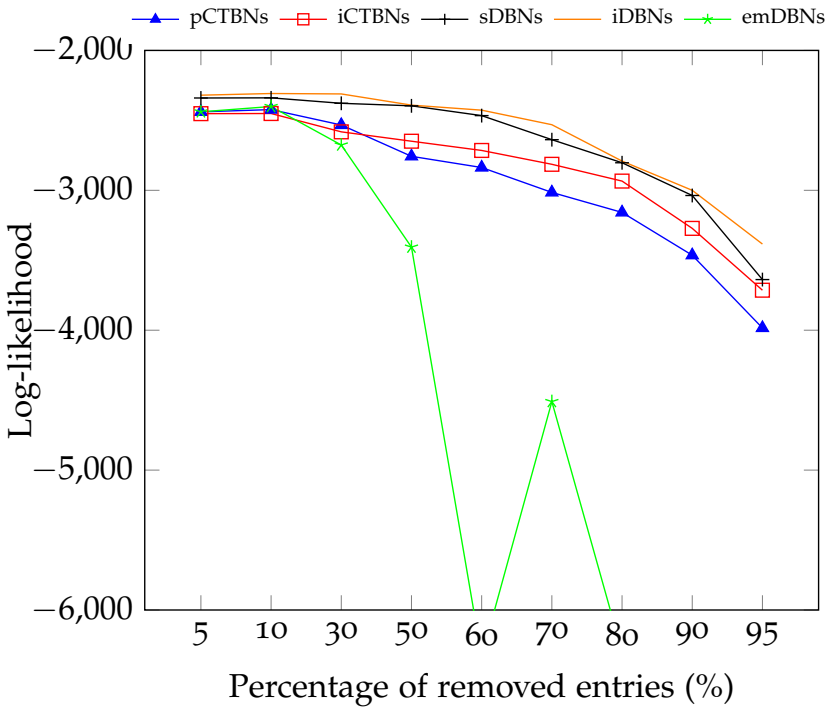
(a)

	Observation rates (day(s))									
	1	2	3	4	5	6	7	14	21	28
DBNs vs iCTBNs	0.31	0.24	0.25	0.26	0.18	0.08	0.14	0.09	0.10	0.10
DBNs vs pCTBNs	0.06	0.10	0.42	0.14	0.82	0.74	0.65	0.11	0.10	0.03
pCTBNs vs iCTBNs	0.24	0.86	0.52	0.56	0.08	0.12	0.15	0.01	0.00	0.01

(b)

Fig. 5.8: Performance of CTBNs and DBNs for regular time series D_{REG} where observations are made at different rates. (a):log-likelihood of CTBNs and DBNs; (b): p-values based on the paired t-test with bold text indicating a higher log-likelihood and with underline indicating a significant difference.

between DBNs and pCTBNs is not always statistical significant (in the experiments, the significance is reached only when the observation rate is 28 days).



(a)

	Percentage of removed entries								
	5%	10%	30%	50%	60%	70%	80%	90%	95%
iDBNs vs emDBNs	0.05	0.12	0.02	0.01	0.03	0.00	0.00	0.00	0.00
sDBNs vs emDBNs	0.10	0.33	0.05	0.01	0.03	0.00	0.00	0.00	0.00
iCTBNs vs emDBNs	0.87	0.52	0.58	0.02	0.04	0.00	0.00	0.00	0.00
pCTBNs vs emDBNs	1.00	0.52	0.24	0.03	0.04	0.00	0.00	0.00	0.00
iDBNs vs sDBNs	0.26	0.18	0.31	0.97	0.78	0.63	0.97	0.91	0.59
sDBNs vs iCTBNs	0.13	0.22	0.20	0.20	0.29	0.54	0.69	0.58	0.90
sDBNs vs pCTBNs	0.01	0.12	0.15	0.01	0.01	0.01	0.12	0.00	0.22
iDBNs vs iCTBNs	0.05	0.06	0.02	0.02	0.01	0.02	0.15	0.03	0.09
iDBNs vs pCTBNs	0.01	0.02	0.02	0.02	0.02	0.05	0.20	0.13	0.09
iCTBNs vs pCTBNs	0.86	0.71	0.61	0.55	0.57	0.47	0.51	0.61	0.59

(b)

Fig. 5.9: Performance of CTBNs and DBNs for irregular time series D_{MCAR} where observations are made completely at random in time. (a): log-likelihood of CTBNs and DBNs; (b): p-values based on the paired t-test with bold text indicating a higher overall log-likelihood and with underline indicating a significant difference.

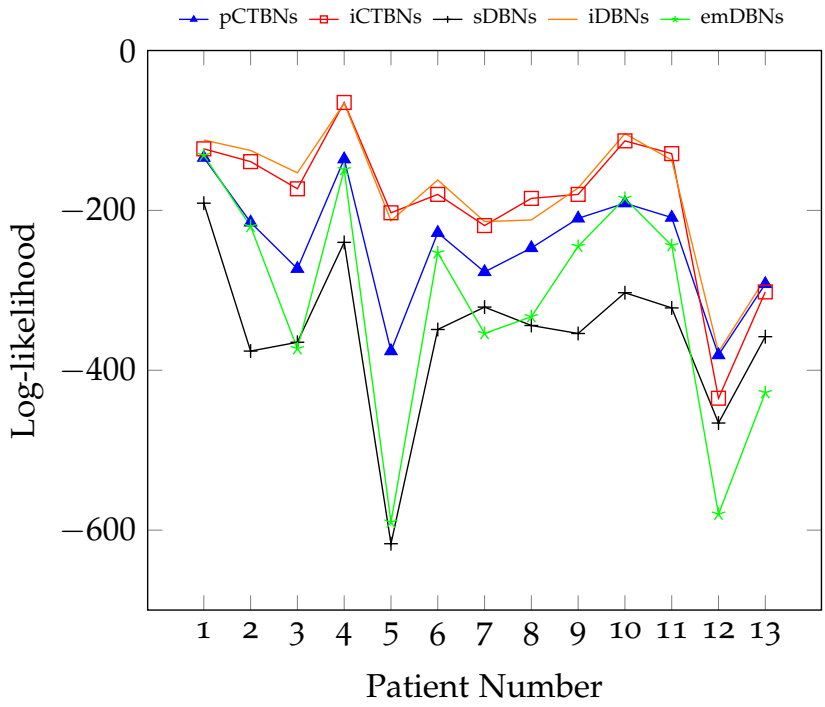
OBSERVATIONS MADE COMPLETELY AT RANDOM Given irregular time series with observations made completely at random in time, we study the impact of the data removal on the performance of DBNs and CTBNs. The log-likelihood of these models learned from a number of time series with a wide range of data removals is summarized in Fig. 5.9. Data removal is represented by a given percentage of removal of entries from the regular time series D_L.

Overall, the results suggest a positive correlation between the performance of DBNs and CTBNs with the data removal. When we take a closer look, the results indicate that it is significantly more difficult for emDBNs to capture the underlying dynamics when more than half of the entries are removed from D_L (see the p-value for emDBNs in Fig. 5.9b). In particular, when 60% of the entries are removed, the drop in the performance of emDBNs also indicates that it is most likely that the EM search does not reach a global optimum. In addition, DBNs using the imputation method have a significantly higher performance than CTBNs for the most of the removal percentages, irrespective of the evidence type.

When studying the behavior of DBNs alone, we also find significant differences of the performance of DBNs using the three distinct ways of interpreting time series. In particular, we find that the performance of DBNs using the EM algorithm declines at an increasing speed, while the performance of the other two DBNs declines relatively slowly. In DBNs, a consistent higher performance is also achieved by exploiting the imputation technique rather than simply discarding time stamps in irregular time series, although the difference is statistically insignificant based on the results on our synthetic datasets.

Now we switch our attention to the behavior of CTBNs. The results indicate that the learned CTBNs using interval evidence are better at capturing the underlying dynamics than these learned using point evidence, while the difference is not statistically significant.

OBSERVATIONS MADE AT RANDOM For irregular time series where observations are made at random in time, the performance of DBNs and CTBNs learned from the reduced irregular time series D_{MAR} is shown in Fig. 5.10. Unlike the previous two cases, the results from time series D_{MAR} are presented at the patient level. In general, the iDBNs learned by filling in missing values at discrete time points and CTBNs learned using interval evidence perform best. For the other two DBNs, however,



(a)

iDBNs vs emDBNs	1.6e-04
emDBNs vs sDBNs	9.6e-02
iCTBNs vs emDBNs	3.5e-04
pCTBNs vs emDBNs	5.1e-03
iDBNs vs sDBNs	1.2e-05
iCTBNs vs sDBNs	4.8e-05
pCTBNs vs sDBNs	4.8e-06
iDBNs vs iCTBNs	1.3e-01
iDBNs vs pCTBNs	2.4e-04
iCTBNs vs pCTBNs	3.5e-03

(b)

Fig. 5.10: Performance of CTBNs and DBNs for irregular time series D_{MAR} where observations are made at random in time. (a):log-likelihood of CTBNs and DBNs; (b): p-values based on paired t-test with bold text indicating a higher overall log-likelihood and with underline indicating a significant difference.

the CTBNs with point evidence perform significantly better. When the focus is on the performance of iDBNs, the performance of these models appear to be rather vulnerable to the reduction of non-transition entries, i.e., when there is no transition between two consecutive entries. This is illustrated by their significantly worse performance on the time series D_{MAR} than on the time series D_{REG} with the observation rate set to one day. Moving to the performance at patient level, we also find that the performance of modeling individual patient using all the temporal models is similar, whereas the log-likelihood is much lower when evaluating on patient 5 and 12.

5.4.2.2 *Results for the ACCESS Dataset*

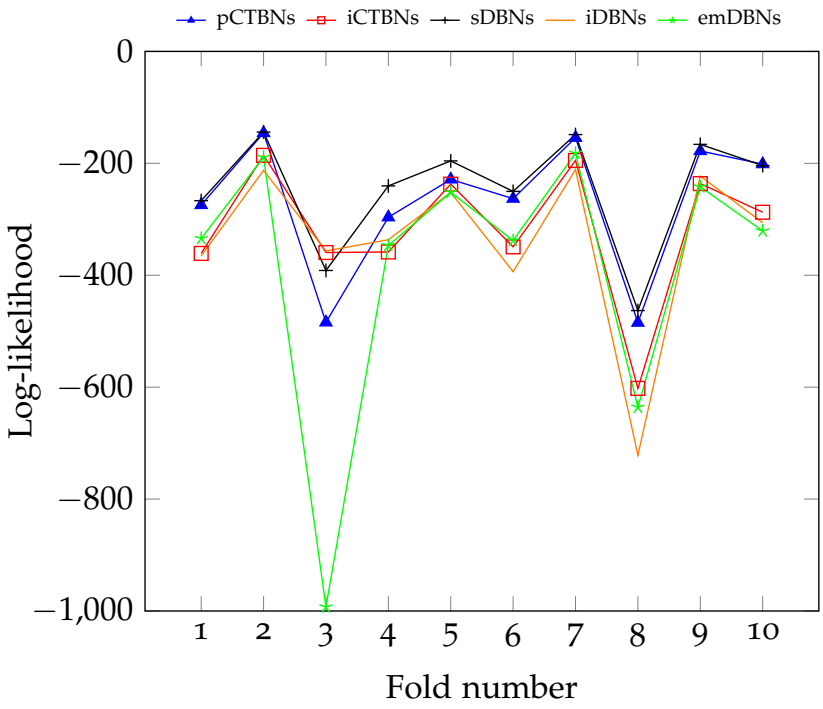
Next, we investigate the performance of DBNs and CTBNs on the real-world irregular clinical time-series D_A; the results are summarized in Fig. 5.11. In general, the results show that sDBNs and pCTBNs have the best performance. By contrast, learning sDBNs performs poorly on the irregular time series D_{MAR}, as shown in Fig. 5.10. The difference can be explained by noting that the time series D_A differs from the time series D_{MAR} mainly in the number of non-transition entries. More specifically, the number of non-transitions in D_{MAR} is zero since it is generated by removing all the transitions from the time series D_L.

To further validate that the difference in the number of non-transitions may be the explanation for the better performance of sDBNS on the time series D_{MAR}, we considered its performance on the irregular time series D_{L2A}. Indeed, we do see an improved performance of sDBNs with an increasing number of non-transition entries in the time series D_{L2A} (see the increased log-likelihood of sDBNs in Fig. 5.12).

Therefore, the results from the two irregular time series D_{L2A} and D_A suggest that treating irregular time series as a sequence by leaving out time stamps may be sufficient to capture the transition probabilities of COPD symptoms. However, the pCTBNs provide a competitive and attractive alternative to these simple DBN models.

5.4.3 *Discussion*

REGULAR DATA WITH DIFFERENT OBSERVATION RATES Given a regular time series, the performance of DBNs and CTBNs deteriorates when the number of transitions in the time series D_{REG} decreases. The



(a)

iDBNs vs emDBNs	5.1e-01
sDBNs vs emDBNs	3.1e-02
iCTBNs vs emDBNs	3.2e-01
pCTBNs vs emDBNs	3.7e-02
sDBNs vs iDBNs	4.0e-03
sDBNs vs iCTBNs	1.4e-03
sDBNs vs pCTBNs	3.1e-02
iDBNs vs iCTBNs	1.3e-01
pCTBNs vs iDBNs	4.7e-02
pCTBNs vs iCTBNs	5.9e-02

(b)

Fig. 5.11: Performance of CTBNs and DBNs for a real-world irregular time series D_A. (a):log-likelihood of CTBNs and DBNs; (b): p-values based on the paired t-test with bold text indicating a higher overall log-likelihood and with underline indicating a significant difference.

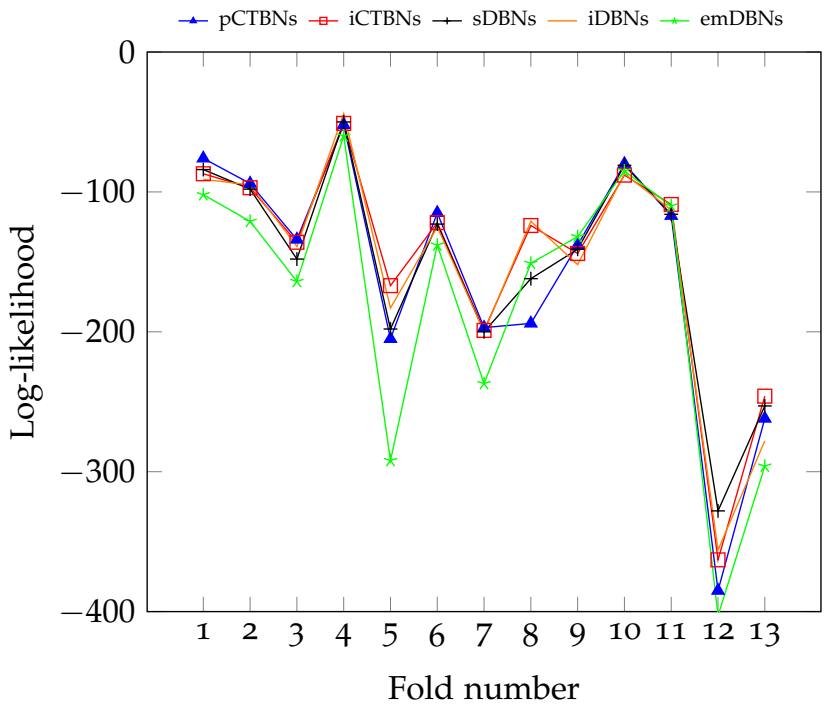


Fig. 5.12: Performance of DBNs and CTBNs on a synthetic time series D_{L2A}, derived from D_L based on D_A.

time series D_{REG} were generated from the time series D_L by removing all transitions that occur within the amount time of a given observation rate. With fewer transitions, the task of learning DBNs becomes more difficult, and restoring the underlying transition probabilities degradates. In addition, it negatively affects the performance of CTBNs using interval evidence, as long time intervals can overestimate the length that a symptom stays in a particular state.

Conversely, increasing the observation rate can be expected to have a less negative impact on learning CTBNs with point evidence, thanks to the presence of intact information at discrete time points in the time series D_{REG}. This is confirmed by the results as shown in Fig. 5.8b: the underlined numbers indicate that CTBNs with point evidence can achieve a significantly higher performance than interval evidence given a sufficiently high observation rate.

The performance difference between DBNs and CTBNs may be due to the ability of DBNs to represent atemporal dependence relative to a given observation rate. In particular, an atemporal dependence can be exploited by DBNs to represent correlations between symptoms that are not evolving over a long time span. For example, in the context of COPD, symptoms may influence each other within days. For a regular time series with larger time granularity (e.g. a week), the correlations within a week can not be captured by temporal dependences. Given that CTBNs do not contain atemporal dependences, we speculate that CTBNs using interval evidence may suffer most from this limitation in expressiveness for regular time series with a large time granularity.

OBSERVATIONS MADE COMPLETELY AT RANDOM The iDBNs gain their advantage over sDBNs and emDBNs by imputing correctly most values of the symptoms at discrete time points. Consider the entries removed from the fragment in Fig. 5.1a, in particular at time 2,9 and 10, standing for normal symptoms and also dominating in the dataset D_L, can be successfully restroed using the imputation method. Discarding time stamps, sDBNs may suffer more from removing entries from the time series D_L, which results in a different probability distribution in the time series D_{MCAR}.

The lower performance of CTBNs with point evidence compared to interval evidence indicates that they may be more vulnerable to information loss than their counterpart with interval evidence. More specifically, the decrease in the number of observations in a time series can

significantly increase the learning search space for CTBNs with point evidence. In contrast, this decrease has a less negative impact on CTBNs with interval evidence. This may be attributed to a better approximation of the duration of the presence and absence of a symptom using interval evidence.

OBSERVATIONS MADE AT RANDOM The much lower log-likelihood for patient 5 is partly because there are significantly more observations for the patient in comparison to the others, as shown in Fig. 5.2. Having more observations often implies a lower log-likelihood. Moreover, patient 12 also differs from the other patients by experiencing a fever, having a cough after the presence of sputum volume, and having more often an abnormal level of oxygen saturation. Such a variance may not be captured by a model that was also learned from data of the other patients, leading to a model that fits less well. Combined with the small size of the study patient population, the results also indicate that there is a need for the development of personalized clinical models, which can be flexibly adapted to each individual patient's behavior.

5.5 CONCLUSION

The main motivation for this research was the wish to provide an alternative to the medically common way of managing symptom worsening of a chronic disease in terms of the *static* occurence of particular symptoms. As an example we used COPD, worldwide a very common chronic disease that increasingly is managed in the home environment through eHealth technology [42, 97]. Methods to capture symptom *dynamics* with their associated uncertainty was seen as a way to make progress here. However, as we discussed, there are significant challenges with respect to learning from clinical data that is collected in a home environment. In order to gain more insight into the most appropriate modeling technique for clinical time series with observations that are unevenly spaced over time, we have studied the performance of dynamic Bayesian networks and continuous-time Bayesian networks on both synthetic and real-world datasets with the final goal to build a predictive model for COPD. For simple cases, such as regularly-spaced time series, discrete-time Bayesian-network models are appropriate, as one might expect. However, for complex clinical time-series data that motivated this research, the continuous-time models are at least compet-

itive and sometimes better than their discrete-time counterparts. Given that CTBNs also provide more fine-grained predictions over time, they are an attractive alternative to discrete time probabilistic models.

Besides this general conclusion, we also studied the usage of DBNs and CTBNs in more detail. Firstly, we have studied different manners to interpret data with temporal Bayesian networks. We showed that the evidence type has a significant impact on the results in different situations. Secondly, we have considered the impact of the hyperparameters (i.e. imaginary counts) in CTBNs. To the best of our knowledge, we are the first to explore the impact of hyperparameters on learned the models, which again has a significant impact on the results that one can obtain with CTBNs.

Our work also has some limitations. First, we only investigate one possible missingness case where variables are either all missing or all observed at each time point. If some random variables are missing and some others are observed at particular points in time, then this would create an additional complexity for learning and reasoning with temporal Bayesian networks.

Besides other types of missing data, we believe that there are a number of other interesting questions to investigate in the future. First of all, choosing the length of the interval when using interval evidence could have a significant impact on the results, as a larger interval provides a stronger bias. While in this paper we have chosen a fairly arbitrary interval determined by the time granularity of the data, it might be more sensible to assist the learning process with domain knowledge about the maximum length of the interval, e.g. a week. Second, it is an intriguing but challenging task to provide theoretical evidence to support the superiority of one method over the other under certain missingness conditions. Third, inspired by data with different time granularities, there is still some room to study the difference between short-term and long-term dynamics for COPD symptoms. Models with multiple time scales, such as provided by continuous-time models, can provide information about the evolution of symptoms from a different perspective. For the final purpose of COPD prediction, this is one of the attractive aspects of CTBNs compared to discrete-time models with a single time granularity.

To conclude, in this paper, the capability of DBNs and CTBNs to handle time series data was studied with a specific medical problem in mind, i.e., COPD patient management. However, the principles concern-

ing their use in practice also apply to other real-world problems where multivariate time series are involved. For example, having an appropriate interpretation of time series is crucial for choosing between DBNs and CTBNs for a given problem. Through close collaboration with domain experts, we can use domain knowledge to better determine the factors for preferring one technique over another. In the medical domain, it may be of clinical interest to have a more fine-grained prediction for the evolution of a disease. In that case, CTBNs are a powerful modeling tool to deal with such predictions. In addition, it is better to use domain knowledge to choose between point and interval evidence in CTBNs. Finally, domain knowledge about the time that a variable would typically stay in a particular state can assist the model learning process.

6

MAKING CONTINUOUS TIME BAYESIAN NETWORKS MORE FLEXIBLE

The time duration in continuous time Bayesian networks (CTBNs), i.e., the time that a variable stays in a state until it transitions to another state, follows an exponential distribution. The exponential distribution is widely applied to describe the waiting time between events in a Poisson process, which describes the number of events in one unit of time. The exponential distribution is parameterized by a single rate and has mode zero. This implies that it is probable that the next event occurs shortly after the last event, which means that we cannot easily model delays between events. Therefore, for biological processes, the exponential distribution is not always realistic. For example, the adaptive immune system typically responds to a viral infection after a few days if it has not encountered the pathogen before. As a consequence, one would not expect that the mode for the duration of an infection is zero.

In this chapter, we make CTBNs more flexible by supporting a richer time duration distribution, which is called the hypoexponential distribution. A hypoexponential distribution is a distribution where its mode can take a positive value. We provide a new extension of CTBNs where the time duration is hypoexponentially distributed. The proposed CTBNs are better-suited for modeling temporal processes, in particular those where the most probable time between events is non-zero.

6.1 INTRODUCTION

Describing waiting time, the time between one event and its consecutive events, is an important part of modeling real-world problems involving time. For example, a question of clinical interest concerning viral infections is how much time it takes for an individual to become infected after contact with an infected other individual. Waiting time, however, is usually described by an exponential distribution in CTBNs, where

the most probable time between events is assumed to be zero. This assumption is not realistic for biological processes. Clearly, an individual can not immediately get a viral infection or recover immediately from a viral disease. Thus, there is a need to seek more versatile distributions to describe non-zero most probable waiting time to handle realistic real-world problems.

The limitation of exponential distributions in CTBNs was first described by Nodelman and Horvitz [69], and they proposed to extend the methodology by replacing the exponential distributions with Erlang distributions. Subsequently, Gopalratnam et al. [29] proposed to use Erlang-Coxian distributions for the duration distribution. These two distributions, Erlang and Coxian distributions, are two special cases of a more general distribution called a phase-type distribution. A phase-type distribution is a probability distribution constructed by a mixture of exponential distributions. More importantly, a phase-type distribution can provide an appropriate approximation of any positive-valued distribution, and provide a greater variety of rates than the exponential distribution.

More recently, Nodelman et al. [73] described two approaches to describe phase-type duration distributions in CTBNs. The phase-type distribution is described by a Markov chain over variables with exponential time distributions. In CTBNs, such a distribution can be represented by the structure of parameters using additional hidden states of random variables, which is called the *direct approach*. The direct method can explicitly give us the transitions between exponentially distributed variables in the Markov chain representing the phase-type distribution. Alternatively, Nodelman suggested another more elegant and cleaner approach by using additional hidden variables [73], which is called the *hidden approach*. Using the hidden approach, a phase-type distributed variable carries a clear contextual meaning, which is lacking in the direct approach. Nodelman et al. also observed that *many complex duration distributions expressible by the direct method can not be expressed by using hidden variables*. The parameters corresponding to a simultaneous change in the states of a variable and its hidden variable are restricted to zeros in the hidden approach, whereas there is no such restriction in the direct approach.

In addition to the lack of contextual meaning, a number of problems arise when using the direct approach in both learning and inference. From a learning perspective, there are many challenges of learning from

data, in particular due to the lack of an appropriate interpretation of evidence. This stems from the fact that we can only observe the current *state* of a variable, but not its associated hidden states. This makes it more challenging to learn from data with missing values. From an inference point of view, the query whether a variable stays in a given state has to be represented by a disjunction of its associated hidden states, which is not always appropriate, in particular for describing interval evidence. Interval evidence takes the form of a variable staying in a given state during a given time interval. Although the state of the variable does not change in the time interval, there are myriad interpretations of the evidence by using the combinations of hidden states associated to the state of the variable. For example, we can choose an arbitrary hidden state for the entire time interval or any arbitrary combinations of hidden states at each time point in the given time interval.

The hidden approach is a cleaner and more elegant representation where the contextual meaning of a phase-type variable and its phase-type time distribution are separated by employing an additional hidden variable. In other words, the phase-type variable itself still carries the meaning of the problem in a given domain, while its associated phase-type distribution is controlled by an additional hidden variable which allows the phase-type variable to have a phase-type time duration distribution. Such a separation is particularly useful for many real-world systems, such as medical diagnostic systems, where the contextual meaning of a phase-type variable is important to understand the underlying dynamics. In addition, the hidden approach has the potential to gain computational advantage by factorizing parameters of a phase-type variable in the direct approach.

In this chapter, we propose a new extension of CTBNs by introducing a more flexible distribution using the hidden approach. First, we show that the hidden approach can actually be used to represent a large and useful subclass of phase-type distributions, known as *hypoexponential distributions*, with the resulting models called hypoexponential continuous time Bayesian networks, or HCTBNs for short. The hypoexponential distributions significantly generalize the existing exponential distribution. Second, we give precise conditions on the CTBN graphs and discuss the exact constraints on the parameter structure for representing hypoexponential distributions using the hidden method, which was not discussed by Nodelman [73]. Third, we investigate the performance of the direct and hidden approaches to estimate parameters of HCTBNs from com-

plete data. The direct approach is expected to be more computationally efficient at the expense of the quality of learned models than the hidden approach. To compare these two methods, a number of measures are used, such as the learning speed in terms of computation time and the quality of learned models in terms of log-likelihood.

6.2 MOTIVATING EXAMPLE

To illustrate the usefulness of the proposed theory, we reconsider the COPD example, previously introduced in Example 2.1 of Chapter 2. Obtaining some insight into the evolution of a chronic disease is an important aspect of disease management, as often the disease will not disappear. In the context of COPD, it is of particular importance to study the effect of a viral infection on lung function, as a viral infection is a major risk factors of a COPD exacerbation, i.e., the disease gets worse. Having a better understanding of the incubation time, i.e., the time for a COPD patient to get a viral infection, and the recovery time, i.e., the time that a COPD patient will recover from a worsening in lung function, is important for standardizing and optimizing the duration of medical treatment. As an example to illustrate the potential of the developed methods, we consider in this chapter the modeling of incubation and recovery time of a viral infection of a COPD patient.

There have been many previous efforts to model the incubation time of viral infections. For example, according to Bailey [5] and Gough [31] the incubation time duration can not be approximated by an exponential distribution, as the mode (the maximum value of the associated density) of the underlying distribution can be far away from zero, whereas the mode of the exponential distribution is actually zero. One example of such non-zero mode distributions is illustrated in Fig. 6.1, which is generated by simulating the incubation time distribution from empirical data from [5, 31]. It is obvious that such a distribution can not be well-captured by the exponential distribution. Exponentially distributed events tend to occur close together, which is not an accurate model for the incubation time of viral infectious diseases. Incubation may take a certain amount of time, due to the fact that virus particles have to be replicated before they are present in sufficient quantity to affect the body.

For COPD patients, the lung function can deteriorate over time, characterized by worsening lung-related symptoms, such as dyspnea. In clin-

ical practice, a peak flow meter is often used as a tool to measure lung function. In the literature, Seemungal et al. [86] have shown that it takes one to two weeks for the majority of moderate to severe COPD patients to recover from their worsening lung function in terms of peak flow. This implies that the mode of the underlying time distribution for recovering from decreased lung function is non-zero.

The examples of incubation time and lung function recovery both indicate that the exponential distribution can not offer a satisfactory representation for many disease processes. As the exponential distribution is a limitation of present CTBNs, we seek more versatile and flexible distributions to handle more complicated real-world problems.

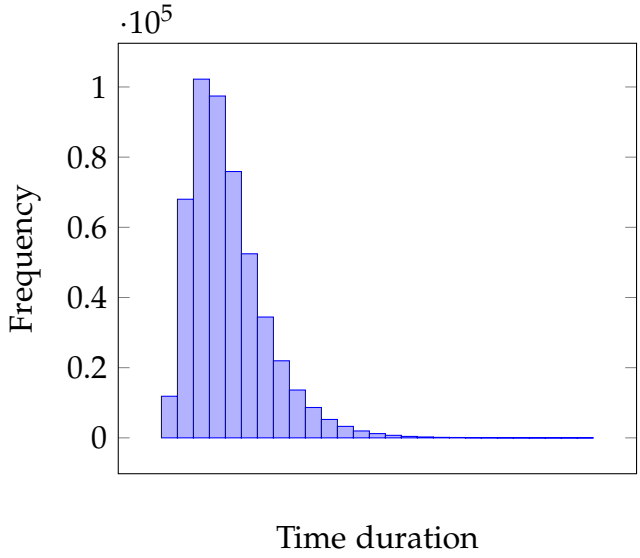


Fig. 6.1: A time distribution generated according to the incubation time in the work by Bailey [5] and Gough [31] based on non-zero mode.

6.3 PRELIMINARIES

Before introducing the hypoexponential distribution, we will first briefly introduce the phase-type distribution, which is the generalization of the hypoexponential distribution.

A phase-type distribution is a probability distribution that is a mixture of exponential distributions. The distribution is represented by a random variable describing the time that a finite-state absorbing continuous time Markov chain reaches its only absorbing state, i.e., a state that

once entered cannot be left. Each state of the Markov chain represents a stochastic process with an exponential time distribution. Before the chain reaches its absorbing state, it moves through its transient states, i.e., a state that once it is reached, there is a non-zero probability that it will never return to the state [98]. In other words, the probability of returning to the state is less than one after visiting it.

Since an n-order phase-type distribution is described by a Markov chain $\{X_t \mid t \in \mathbb{R}_0^+\}$ with n transient states $1, 2, \ldots, n$ and one absorbing state $n+1$, its parameters are fully specified by an initial distribution vector $\mathbf{p} = [p_1, \ldots, p_n]^T$ ($\mathbf{p}^T$ is the transpose of $\mathbf{p}$), $p_i = P(X_0 = i)$, $i \in \{1, \ldots, n\}$, with $\sum_{i=1}^{n} p_i = 1$ (the probability for the absorbing state is zero, i.e., $P(X_0 = n+1) = 0$), and an intensity matrix Q as previously introduced in Section 2.4. The intensity matrix Q differs from general intensity matrices by the fact that transitioning away from the absorbing state has zero intensity. More specifically, the intensity matrix Q is defined as follows:

$$Q = \begin{pmatrix} A & \mathbf{v} \\ \mathbf{0}^T & 0 \end{pmatrix}$$

where A is an $n \times n$ dimensional (square) matrix, specifying the intensities for transitioning between transient states, $\mathbf{v}$ is a column vector where v_i is the intensity leaving transient state i to the absorbing state, and $\mathbf{0}^T$ is a zero row vector of dimension n. The matrix A is called the phase-type generator, and n is called the order of the phase-type distribution. Every row in matrix Q sums to zero, from which it follows that $\mathbf{v} = -A\mathbf{e}$, where $\mathbf{e} = [1, 1, \ldots, 1]^T$ is an n-dimensional column vector of ones.

The density function for the phase-type distribution is defined as:

$$f(t) = \mathbf{p}^T \exp(At) \cdot -A\mathbf{e} = \mathbf{p}^T \exp(At)\mathbf{v}$$

and the distribution function is:

$$\begin{aligned} F(t) &= \int_0^t f(s)\mathrm{d}\, s \\ &= \int_0^t \mathbf{p}^T \exp(As) \cdot -A\mathbf{e}\, \mathrm{d}\, s \\ &= \left[-\mathbf{p}^T \exp(As)\mathbf{e}\right]_0^t \\ &= -\mathbf{p}^T \exp(At)\mathbf{e} + \mathbf{p}^T \exp \mathbf{0} \\ &= -\mathbf{p}^T \exp(At)\mathbf{e} + \mathbf{p}^T\mathbf{e} \\ &= 1 - \mathbf{p}^T \exp(At)\mathbf{e} \end{aligned}$$

For an n-order phase-type distribution, its associated continuous time Markov chain can be graphically represented by a state transition diagram. The diagram is a convenient graphical representation by specifying the initial probabilities $\mathbf{p}$, i.e., the distribution over the all the transient states when the chain starts, the rates between the transient states, and the exit rates, i.e., the rate for a transient state entering the absorbing state. More details about phase-type distributions can be found in [9, 98].

In this chapter, we mainly focus on the hypoexponential distribution, also known as *generalized Erlang distribution*. It is a rich and flexible phase-type distribution. For a hypoexponential distribution, a transient state is only allowed to enter its consecutive transient state or the absorbing state in the corresponding Markov chain. In addition, it starts only in one of the transient states and traverses all the other transient states until it reaches the absorbing state. The n-order hypoexponential distribution is graphically represented by a Markov chain with its state diagram as shown in Fig. 6.2. The diagram asserts that the chain enters transient state 1 with probability one and enters the absorbing state with rate λ_n. The specification of the hypoexponential distribution consists of the vector of initial probabilities $\mathbf{p} = [1, 0, \ldots, 0]^T$, the matrix A:

$$A = \begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & \cdots & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -\lambda_n \end{pmatrix}$$

and $\mathbf{v} = [0, 0, \ldots, \lambda_n]^T$.

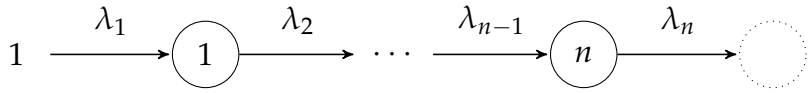


Fig. 6.2: A state transition diagram for an n-order hypoexponential distribution. A solid node indicates a transient state and a dashed node indicates an absorbing state. The number '1' at the left-hand side denotes the initial probability of entering a state. Arcs stand for temporal dependences, same for all the arcs in the remainder of this chapter.

6.4 HIDDEN CONTINUOUS TIME BAYESIAN NETWORKS

In this section, we define a new extension of CTBNs, which we call hidden continuous time Bayesian networks, abbreviated to HCTBNs, where the time duration follows a more versatile distribution. In HCTBNs, the time duration, i.e., the time that a variable stays in a state until a transition occurs, is described by the hypoexponential distribution. An auxiliary hidden variable is introduced to a hypoexponential variable to model the hypoexponential distribution. These auxiliary hidden variables only serve as mediators and convey no contextual meanings. In HCTBNs, variables are categorized into three groups, hypoexponentially distributed variables, their corresponding auxiliary hidden variables, and exponentially distributed variables. For a hypoexponential variable, the time duration follows a hypoexponenital distribution while it can only take two possible values, indicated by 1 and 2,

In the remainder of this section, we define an HCTBN in terms of its graphical structure and the structure of its intensity matrices. More specifically, a set of constraints on intensity matrices are imposed, in particular for hypoexponential variables and auxiliary hidden variables. In the remainder of this chapter, hypoexponential variables are assumed to be binary-valued.

6.4.1 *Structure*

First, we define the graphical structure associated to an HCTBN, with labeled nodes $\mathbf{X}$ for binary hypoexponential variables, labeled nodes $\mathbf{H}$ for auxiliary hidden variables and labeled nodes $\mathbf{Y}$ for exponential variables. The set of possible values of a binary variable X are denoted

by $\{1, 2\}$ and of an n-valued hidden variable or exponential variable by $\{1, 2, \ldots, n\}$.

Definition 6.1 (HCTBN Graph)**.** *An HCTBN graph is a node-labeled graph defined by a tuple* $G = (\mathbf{V}, \mathbf{E}, b, l)$*, where* $\mathbf{V} = \mathbf{X} \cup \mathbf{H} \cup \mathbf{Y}$ *denotes a set of nodes where* $\mathbf{X}$*,* $\mathbf{H}$ *and* $\mathbf{Y}$ *are mutually disjoint,* $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ *a set of arcs on* $\mathbf{V}$*,* b *a bijective function* $b : \mathbf{X} \to \mathbf{H}$*, and* l *a label function such that* $l(\mathbf{X}) =$ *hypoexponential,* $l(\mathbf{H}) =$ *hidden, and* $l(\mathbf{Y}) =$ *exponential. Furthermore, the graph* G *has the following properties:*

1. *For any* $X \in \mathbf{X}$*,* $b(X) \to X \in \mathbf{E}$ *and* $X \to b(X) \in \mathbf{E}$*;*
2. *For any* $Z \in \mathbf{Y} \cup \mathbf{X}$*,* $Z \to X \in \mathbf{E}$ *iff* $Z \to b(X) \in \mathbf{E}$*,* $X \in \mathbf{X}$*;*
3. *For any* $H \in \mathbf{H}$*,* $H \to Y \notin \mathbf{E}$*;*
4. *For any* $H, H' \in \mathbf{H}$*,* $H \to H' \notin \mathbf{E}$*.*

Definition 6.1 states that for any $X \in \mathbf{X}$, there is exactly one associated hidden variable $H, H = b(X)$. This also implies that in an HCTBN $\mathbf{X}$ and $\mathbf{H}$ have the same cardinality. In addition, there are three main restrictions on the arcs in the HCTBN graph. First, Property 1 asserts that node X is connected to its associated hidden node $b(X)$ by a bidirected arc. Second, Property 2 asserts that the associated hidden node $b(X)$ has an exponential node $Y, Y \in \mathbf{Y}$, as a parent if and only if Y is also a parent of X. Of course, Y can be a parent of more than one hypoexponential nodes, and thus be a parent of more than one hidden nodes. For example, we may have an exponential node Y be a parent of two distinct hypoexponential variables X, X', i.e., $Y \to X$ and $Y \to X'$. According to Definition 6.1, node Y is also a parent of the associated hidden nodes $b(X)$ and $b(X')$, i.e., we also have $Y \to b(X)$ and $Y \to b(X')$ in the graph. The graph properties imply that for each hypoexponential node X, the node has the same number of parents as its associated hidden node $b(X)$. Thus, the number of parameters for the hidden node $b(X)$ grows exponentially with the number of parents of node X. Third, Property 3-4 state that a hypoexponential node X is the only child for its associated hidden node $b(X)$. This also implies that there are no direct connections between any two distinct hidden nodes. Property 1-4 make sure that a hidden variable is only used as an auxiliary variable to describe the hypoexponential distribution for a hypoexponential variable.

Example 6.1

Consider two simple HCTBN graphs where we only have one single hypoexpoential node X and its corresponding hidden node $b(X) = H$, i.e., $\mathbf{X} = \{X\}, \mathbf{H} = \{H\}$, as given in Fig. 6.3. In the first graph, we have no exponential nodes, i.e., $\mathbf{Y} = \emptyset$. In the second graph, we have one exponential node Y, i.e., $\mathbf{Y} = \{Y\}$, and node Y is a parent of the hypoexponential node X, and thus a parent of the hidden node H. In both graphs, nodes X and H are connected by a bidirected arc.

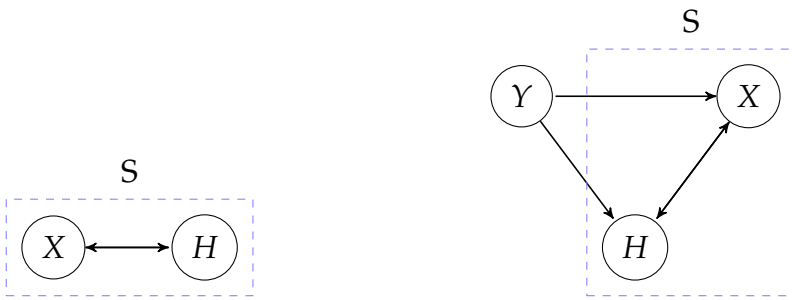


Fig. 6.3: Two HCTBN graphs with a single hypoexponential node X and a single hidden node H, i.e., $\mathbf{X} = \{X\}, \mathbf{H} = \{H\}$, where X has no exponential variables as children. (a) $\mathbf{Y} = \emptyset$, $\pi(X) = \{H\}$ and $\pi(H) = \{X\}$; (b) $\mathbf{Y} = \{Y\}, \pi(X) = \{H, Y\}$ and $\pi(H) = \{X, Y\}$. Arcs stand for temporal dependences, same for all the arcs for the HCTBNs in the remainder of this chapter.

Now we can also describe the time duration for infection I and lung function LF in the COPD network with a more versatile distribution by modeling them as hypoexponential variables in an HCTBN.

Example 6.2

For the COPD problem, the eight variables are categorized into three groups, hypoexponential variables $\mathbf{X} = \{I, LF\}$, exponential variables $\mathbf{Y} = \{V, C, D, W\}$ and hidden variables $\mathbf{H} = \{H_1, H_2\}$ with $b(I) = H_1$ and $b(LF) = H_2$. The corresponding HCTBN graph is depicted in Fig. 6.4. Hypoexponential nodes I and LF have hidden nodes H_1 and H_2 as parents, respectively. In addition, hypoexponential node LF has the hypoexponential variable I as a parent, thus its corresponding hidden node H_2 also has an incoming arc from node I. Furthermore, all the exponential variables are either a child of the hypoexponential node I

or V or LF. It is important to mention that hidden nodes H_1 and H_2 are not directly connected with the exponential variables.

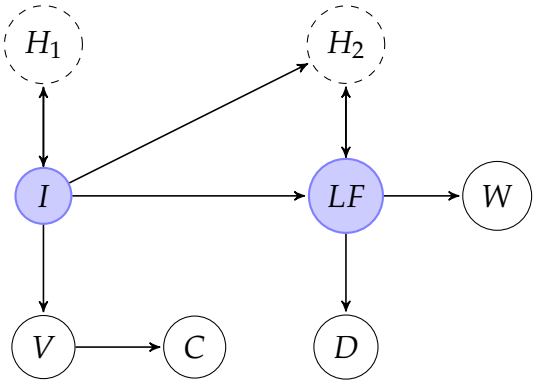


Fig. 6.4: An HCTBN graph for COPD problem. I = infection, V = sputum volume, C = cough, LF = lung function, D = dyspnea, W = wheeze, and H_1 and H_2 are hidden nodes corresponding to I and LF, respectively. Dashed nodes correspond to hidden variables $\mathbf{H}$, solid blue nodes to hypoexponential variables $\mathbf{X}$, and solid nodes to the rest nodes $\mathbf{Y}$.

6.4.2 *Model Definition*

Now we give a formal definition of an HCTBN in terms of both its graph and the structure of its intensity matrices for variables in the HCTBN, in particular for hypoexponential variables and auxiliary hidden variables.

Definition 6.2 (Hidden Continuous Time Bayesian Networks (HCTBNs))**.** *A hidden continuous time Bayesian network (HCTBN) is a triple* $\mathcal{N} = (G, \mathbf{Q}, P_0)$ *with the HCTBN graph* G *as defined in Definition 6.1. In addition,* $\mathbf{Q}$ *is a set of conditional intensity matrices and* P_0 *is the initial distribution for the variables associated to the nodes in the graph* G*. For each* $X \in \mathbf{X}$ *with* n_X*-order hypoexponential distribution and* $H = b(X)$*,* $n_X \in \mathbb{N}, n_X \geq 2$*, we have either* $P_0(X = 1, H = 1) = 1$ *or* $P_0(X = 2, H = n_X) = 1$*, and for each configuration* $\mathbf{u}$ *of the parents* $\mathbf{U}$ *of variable* X*, with* $\mathbf{U} = \pi(X) \setminus \{H\}$*, intensity matrices for variable* X *and* H *are given in Fig. 6.5.*

In Definition 6.2, there are two restrictions on the intensity matrices of a hypoexponential variable X and its associated n_X-valued hidden

variable $H, H = b(X)$. First, the hidden variable H is only allowed to transition from state i to its previous state $i-1$ or next state $i+1$ and it stops at either state 1 or state n_X, after it visits all the other $n_X - 1$ states. The state that variable H transitions to depends on the value of X. For example, when variable X is in state 1, variable H reaches state n_X after it visits the other states in the order $1 \to 2 \to \cdots \to n_X - 1$. Second, variable X is only allowed to make a transition when variable H is in either one of the end states 1 or n_X. These two restrictions impose that variable X does not transition until variable H visits its intermediate states. Once variable H reaches state 1 or n_X, variable X transitions to its other state.

Note that the order for the hypoexponential distribution may differ from one hypoexponential variable to another, hence the use of the subscript X in n_X.

Example 6.3

Consider an HCTBN with its HCTBN graph given in Fig. 6.4. In the model, we consider the intensity matrices for the hypoexponential variable LF and its corresponding hidden variable H_2. Suppose the hypoexponential variable LF has an n_{LF}-order hypoexponential distribution, where $n_{\text{LF}} = 4$, and we have parameters $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, \lambda_4 = 4, \gamma_1 = 5, \gamma_2 = 6, \gamma_3 = 7, \gamma_4 = 8$ when $I = 1$, and $\lambda_1 = 9, \lambda_2 = 10, \lambda_3 = 11, \lambda_4 = 12, \gamma_1 = 13, \gamma_2 = 14, \gamma_3 = 15, \gamma_4 = 16$ when $I = 2$, this gives the intensity matrices for variable LF and H_2 shown in Fig. 6.6.

In addition, hypoexponential variable LF has a number of exponential variables as children and we also consider the structure of the intensity matrix for one of its children D. There are no restrictions on the intensity matrices for variable D:

$$Q_{D|LF=1} = \begin{pmatrix} -29 & 29 \\ 30 & -30 \end{pmatrix} \quad Q_{D|LF=2} = \begin{pmatrix} -31 & 31 \\ 32 & -32 \end{pmatrix}$$

For each hypoexponential variable X and its associated hidden variable H, we can view variable X and H as a whole by amalgamating them into a single variable S, whose state space is the joint state space over X and H. Each state of X now corresponds to a set of instantiations to S. When we amalgamate over the hypoexponential variable X and the hidden variable H, their joint intensity matrix follows a particular structure. Suppose we have the variable order $X < H$ as defined in Section 2.5.2, the joint states over X, H in the intensity matrix are given

$$Q_{\mathrm{H}|X=1,\mathbf{U}=\mathbf{u}} = \begin{pmatrix} -\lambda_1^{\mathbf{u}} & \lambda_1^{\mathbf{u}} & \dots & 0 & 0 \\ 0 & -\lambda_2^{\mathbf{u}} & \lambda_2^{\mathbf{u}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & -\lambda_{n_X-1}^{\mathbf{u}} & \lambda_{n_X-1}^{\mathbf{u}} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$Q_{\mathrm{H}|X=2,\mathbf{U}=\mathbf{u}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \gamma_{n_X-1}^{\mathbf{u}} & -\gamma_{n_X-1}^{\mathbf{u}} & \dots & 0 & 0 \\ 0 & \gamma_{n_X-2}^{\mathbf{u}} & -\gamma_{n_X-2}^{\mathbf{u}} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \gamma_1^{\mathbf{u}} & -\gamma_1^{\mathbf{u}} \end{pmatrix}$$

$$Q_{\mathrm{X}|H=1,\mathbf{U}=\mathbf{u}} = \begin{pmatrix} 0 & 0 \\ \gamma_{n_X}^{\mathbf{u}} & -\gamma_{n_X}^{\mathbf{u}} \end{pmatrix} \quad Q_{\mathrm{X}|H=n_X,\mathbf{U}=\mathbf{u}} = \begin{pmatrix} -\lambda_{n_X}^{\mathbf{u}} & \lambda_{n_X}^{\mathbf{u}} \\ 0 & 0 \end{pmatrix}$$

$$\text{If } n_X \geq 3,\ Q_{\mathrm{X}|H=2:n_X,\mathbf{U}=\mathbf{u}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Fig. 6.5: The structure of intensity matrices for an hypoexponential variable X and its corresponding auxiliary hidden variable H, where $H = b(X)$, given the parents $\mathbf{U} = \pi(X) \setminus \{H\}$.

$$Q_{H_2|LF=1,I=1} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & -3 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad Q_{LF|H_2=1,I=1} = \begin{pmatrix} 0 & 0 \\ 8 & -8 \end{pmatrix}$$

$$Q_{H_2|LF=2,I=1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 7 & -7 & 0 & 0 \\ 0 & 6 & -6 & 0 \\ 0 & 0 & 5 & -5 \end{pmatrix} \qquad Q_{LF|H_2=2:3,I=1} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$Q_{LF|H_2=4,I=1} = \begin{pmatrix} -4 & 4 \\ 0 & 0 \end{pmatrix}$$

$$Q_{H_2|LF=1,I=2} = \begin{pmatrix} -9 & 9 & 0 & 0 \\ 0 & -10 & 10 & 0 \\ 0 & 0 & -11 & 11 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad Q_{LF|H_2=1,I=2} = \begin{pmatrix} 0 & 0 \\ 16 & -16 \end{pmatrix}$$

$$Q_{LF|H_2=2:3,I=2} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$Q_{H_2|LF=2,I=2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 15 & -15 & 0 & 0 \\ 0 & 14 & -14 & 0 \\ 0 & 0 & 13 & -13 \end{pmatrix} \quad Q_{LF|H_2=4,I=2} = \begin{pmatrix} -12 & 12 \\ 0 & 0 \end{pmatrix}$$

Fig. 6.6: Parameters for hypoexponential variable LF and its associated hidden variable H_2 in the HCTBN with its structure given in Fig. 6.4.

$\langle 1,1\rangle, \langle 2,1\rangle, \langle 1,2\rangle, \langle 2,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle, \cdots, \langle 1, n_X - 2\rangle, \langle 2, n_X - 2\rangle, \langle 1, n_X - 1\rangle, \langle 2, n_X - 1\rangle, \langle 1, n_X\rangle, \langle 2, n_X\rangle$.

Proposition 6.4.1. *Let X be an n_X-order hypoexponential variable in an HCTBN. The time that variable X stays in each of its states follows an n_X-order hypoexponential distribution.*

Given the joint intensity matrix of an n_X-order hypoexponential variable X and its associated hidden variable $H, H = b(X)$, now we can reinterpret the time duration of variable X in terms of the joint state over variable X and H. More specifically, the time of variable X staying in a state is then reinterpreted as the absorbing time of a Markov chain with a sequence of joint states over variable H and X where variable X in the joint states remains in the given state. For example, the time that variable X stays in state 1 is thus viewed as the absorbing time of a Markov chain with a sequence of joint states $11, 12, \ldots, 1n_X$, where X always stays in state 1 and the final transition in such a chain is the transition from state $1n_X$ to $2n_X$. As noted, there is no explicit absorbing state. It is clear that such a Markov chain describes an n_X-order hypoexponential distribution. Analogously, we can construct another Markov chain corresponding to state 2 for variable X. Together, we can obtain a single Markov chain that is graphically represented by a cyclic state transition diagram as shown in Fig. 6.7a.

6.5 EQUIVALENT DIRECT MODELS

An important task for any probabilistic graphical model is to estimate parameters from data. As HCTBNs fit naturally into CTBNs framework, the existing learning algorithms can be directly applied to estimate parameters in HCTBNs. Alternatively, HCTBNs can be transformed into their equivalent direct models which have the same time distribution for hypoexponential variables. In this section, we define such equivalent direct models from given HCTBNs. The introduction of these models only serves as a tool to estimate parameters in HCTBNs.

Definition 6.3 (Equivalent Direct Graph)**.** *Let $G = (\mathbf{V}, \mathbf{E})$ be an HCTBN graph with hypoexponential nodes $\mathbf{X}$, hidden nodes $\mathbf{H}$ and exponential nodes $\mathbf{Y}$ and $\mathbf{V} = \mathbf{X} \cup \mathbf{H} \cup \mathbf{Y}$. An equivalent direct graph G' is defined as a graph $G' = (\mathbf{V}', \mathbf{E}')$, with nodes $\mathbf{V}' = \mathbf{X} \cup \mathbf{Y}$ and arcs $\mathbf{E}' = \mathbf{E} \cap (\mathbf{V}' \times \mathbf{V}')$.*

For an HCTBN graph G, an equivalent direct graph G' is a graph that excludes all the hidden nodes $\mathbf{H}$ from the graph G while it includes all

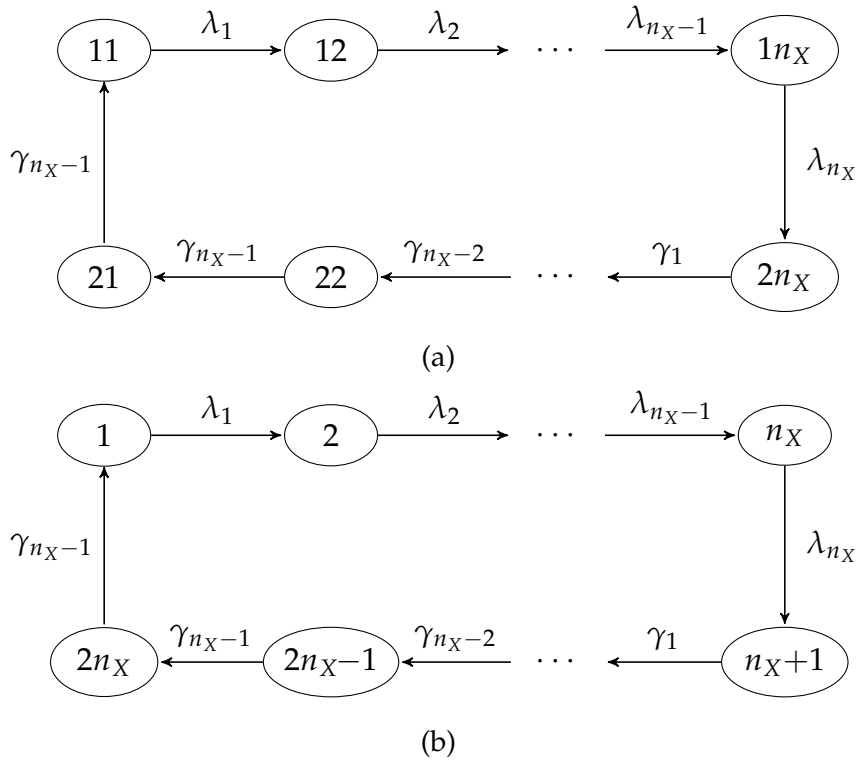


Fig. 6.7: State transition diagram for joint states over an n_X-order hypoexponential variable X and hidden variable H where H is a parent of X in an HCTBN (a) and for states of its extended variable X' in its equivalent Markovian model (b).

the hypoexponential and exponential nodes. Any arcs corresponding to a hidden node are also omitted in graph G'. For example, the hidden nodes are omitted in the equivalent direct model graphs as shown in Fig. 6.8 and Fig. 6.9 with their associated HCTBNs given in Fig. 6.3 and in Fig. 6.4.

Fig. 6.8: Equivalent direct graphs associated to HCTBNs as introduced in Fig. 6.3: (a) $\mathbf{Y} = \pi(X) = \emptyset$; (b) $\mathbf{Y} = \pi(X) = \{Y\}$.

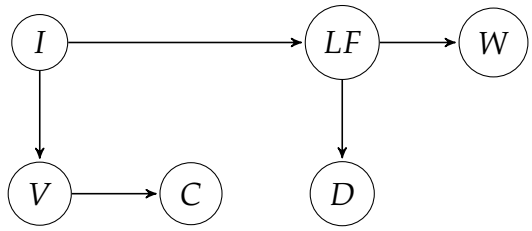


Fig. 6.9: An equivalent direct graph for the HCTBN of the COPD problem as given in Fig. 6.4. I = infection, V = sputum volume, C = cough, LF = lung function, D = dyspnea, W = wheeze.

Definition 6.4 (Equivalent Direct Models)**.** *Let $\mathcal{N}$ be an HCTBN with intensity matrices $\mathbf{Q}$ and graph G. An equivalent direct model $\mathcal{M}$ is defined as a triple $\mathcal{M} = (G', \mathbf{Q}', P'_0)$ where graph $G' = (\mathbf{X}, \mathbf{Y}, \mathbf{E}')$ is defined in Definition 6.3 and $\mathbf{Q}'$ is a set of intensity matrices over the nodes in graph G' and P'_0 is the initial distribution with $P'_0(X = 1) = 1$ or $P'_0(X = n_x + 1) = 1$, for any $X \in \mathbf{X}$.*

In addition, intensity matrices for any variable $Y \in \mathbf{Y}$ satisfy the following conditions:

- If for any $X \in \mathbf{X}, X \notin \pi(Y)$, $Q^{\mathcal{M}}_{Y|\pi(Y)} = Q^{\mathcal{N}}_{Y|\pi(Y)}$ where $Q^{\mathcal{M}}_{Y|\pi(Y)}$ and $Q^{\mathcal{N}}_{Y|\pi(Y)}$ are the intensity matrix for variable Y in $\mathcal{M}$ and $\mathcal{N}$ respectively; otherwise, $Q^{\mathcal{M}}_{Y|\mathbf{K}=\mathbf{k}^{\mathcal{M}},\mathbf{K}'=\mathbf{k}'} = Q^{\mathcal{N}}_{Y|\mathbf{K}=\mathbf{k}^{\mathcal{N}},\mathbf{K}'=\mathbf{k}'}$, where $\mathbf{K} = \pi(Y) \cap \mathbf{X}$ and $\mathbf{K}' = \pi(Y) \cap \mathbf{Y}$, $\mathbf{k}^{\mathcal{M}}$ and $\mathbf{k}^{\mathcal{N}}$ are the values of variables $\mathbf{K}$ in $\mathcal{M}$ and $\mathcal{N}$ respectively, and for any $K \in \mathbf{K}$, if $k^{\mathcal{M}} \in \{1, \cdots n_K\}$, where n_K is the number of possible values for variable K in $\mathcal{M}$, then $k^{\mathcal{N}} = 1$; otherwise $k^{\mathcal{N}} = 2$.

Furthermore, for each variable $X \in \mathbf{X}$, the intensity matrices $Q^{\mathcal{M}}_{X|\pi(X)=\mathbf{u}}$ are defined by re-ordering the states of $Q^{\mathcal{N}}_{\mathrm{XH}|\pi(X)\setminus\{H\}=\mathbf{u}}$ from current indices $[1,\ldots,2n_X]$ to $[1,3,\ldots,2n_X-1,2n_X,\ldots,4,2]$, where $H = b(X)$ in $\mathcal{N}$.

Definition 6.4 gives a general procedure to transform the intensity matrices for variables $\mathbf{Y}$ and $\mathbf{X}$ from an HCTBN $\mathcal{N}$ to its equivalent direct model $\mathcal{M}$. The transformation involves two parts, one part for exponential variables $\mathbf{Y}$ and one part for hypoexponential variables $\mathbf{X}$. For variable $Y \in \mathbf{Y}$, its intensity matrices in $\mathcal{M}$ are simply a copy of those in $\mathcal{N}$ if its parents do not contain any hypoexponential variables. Otherwise, we also need to transform the values of hypoexponential variables from $\mathcal{M}$ to $\mathcal{N}$ before copying intensity matrices from $\mathcal{N}$. For example, variable X is an n_X-order hypoexponential variable and is a parent of Y. Then the intensity matrices $Q^{\mathcal{M}}_{Y|X=1:n_X}$ for variable Y in $\mathcal{M}$ corresponds to $Q^{\mathcal{N}}_{Y|X=1}$ in $\mathcal{N}$, and $Q^{\mathcal{M}}_{Y|X=n_X+1:2n_X}$ to $Q^{\mathcal{N}}_{Y|X=2}$. For a hypoexponential variable X, its intensity matrices in $\mathcal{M}$ are transformed from $\mathcal{N}$ by amalgamating the joint intensity matrix of variable X and its associated hidden variable H and then reordering the resulting joint intensity matrix in a particular order.

For a binary variable X with n_X-order hypoexponential distribution in an HCTBN, the hypoexponential distribution is encoded in the associated n_X-valued hidden variable $H, H = b(X)$. In its equivalent direct model, the hypoexponential distribution is represented by adding additional states to its corresponding variable X'. Thus, the size of the state-space of variable X' grows to $2 * n_X$ in the equivalent direct model from 2 states in the HCTBN.

Definition 6.4 also implies that HCTBNs have the same number of parameters in their equivalent direct models.

Example 6.4

For the HCTBN $\mathcal{N}$ as parameterized in Example 6.3, the structure of its equivalent Markov model $\mathcal{M}$ is given in Fig. 6.9 and we have an 8×8 intensity matrix for variable LF in $\mathcal{M}$ as given in the following:

$$Q_{LF}^{\mathcal{M}} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 7 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & -8 \end{pmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}$$

The size of the intensity matrix of variable LF in $\mathcal{M}$ expands from 2 to 8. In model $\mathcal{M}$, variable LF has eight states and states 1-4 correspond to state 1 for its associated variable in $\mathcal{N}$, and the rest states correspond to state 2.

In addition, we also need to transform the intensity matrices for variable D. Since variable D has the hypoexponential variable LF as a parent, we need to copy intensity matrices from $\mathcal{N}$ corresponding to each state of LF in $\mathcal{M}$, this gives:

$$Q_{D|LF=1:4}^{\mathcal{M}} = \begin{pmatrix} -29 & 29 \\ 30 & -30 \end{pmatrix} \quad Q_{D|LF=5:8}^{\mathcal{M}} = \begin{pmatrix} -31 & 31 \\ 32 & -32 \end{pmatrix}$$

Proposition 6.5.1. *Let X be an n_X-order hypoexponential variable in an HCTBN and X' the extended variable of X in its associated equivalent Markov model. The absorbing time in a Markov chain described by a sequence of states $1, 2, \ldots, n_X$ of variable X' follows the same distribution as the time X stays in state 1, and the absorbing time in a Markov chain described by a sequence of states $n_X + 1, n_X + 2, \ldots, 2n_X$ of variable X' follows the same distribution as the time X stays in state 2.*

Similar to an HCTBN, we can also construct a state transition diagram for its equivalent direct model, as shown in Fig. 6.7b. The time that X stays in state 1 has an n_X-order hypoexponential distribution with rates $\lambda_1, \lambda_2, \ldots, \lambda_{n_X}$. The same distribution can also be represented by a Markov chain of a sequence of states of variable X', $1, 2, \ldots, n_X$. It is similar for the variable X staying in state 2.

6.6 EXPERIMENTS

In the experiments, we investigate two aspects of HCTBNs. First, we investigate whether HCTBNs provide a better approximation than CTBNs when the underlying temporal processes are governed by a hypoexponential time duration distribution. Second, we studied a number of aspects regarding parameter estimation in HCTBNs using the direct and hidden approaches. In both approaches, the EM algorithm is used to estimate parameters as either the hidden variables in HCTBNs are unobserved or the hidden states are unknown in their corresponding direct models. For the EM algorithm used in the methods, we compared the time and the number of iterations until the EM algorithm converges, and the quality of learned models in terms of log-likelihood.

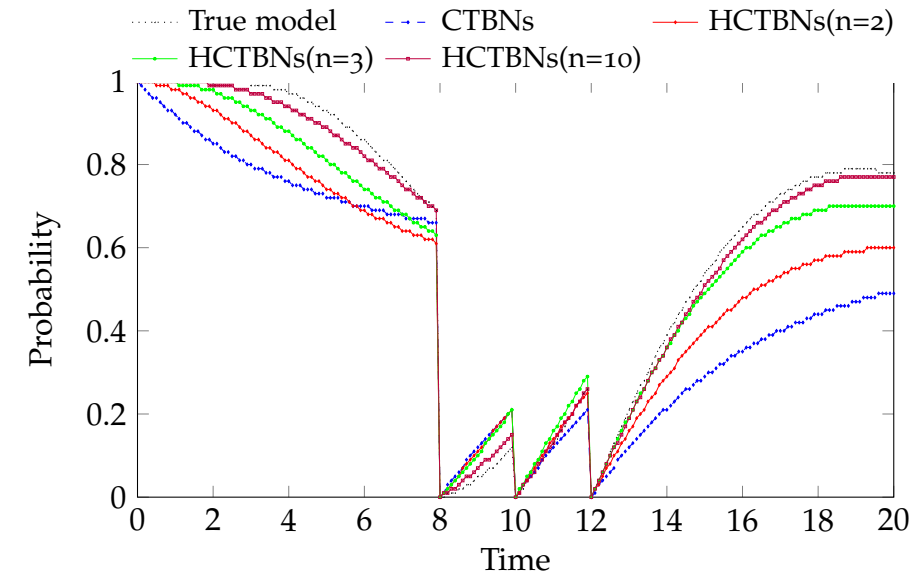
In the experiments, a number of software packages were used to learn parameters for HCTBNs. We used the existing CTBNs learning algorithms in the package *CTBN-RLE*[1] to estimate parameters in HCTBNs directly. For the direct approach, the transformation between a given HCTBN and its equivalent direct representation was implemented in R. We also employed *EMpht*[2] to learn parameters for this direct representation from *right censored data*, i.e., a variable staying in a state for at least a given amount of time. A more detailed discussion about censored data can be found in [29].

For the first part of the experiments, we generated a number of datasets from temporal processes where the time distribution follows a complex hypoexponential distribution, rather than the simple exponential distribution. With respect to learning parameters for HCTBNs, we also considered the impact of the number of states for the hidden variables on the quality of the approximation in the learned HCTBNs. The number of hidden states in the learned models was set to 2,3 and 10 when the underlying hypoexponential distribution has an order 10, and to 3 and 5 when the distribution has order 5.

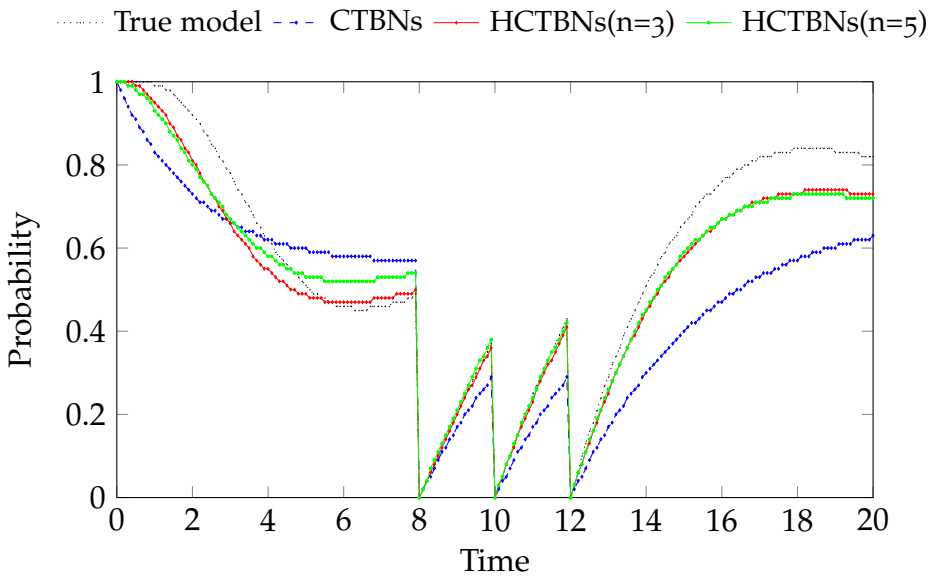
For illustrative purposes, we considered learning parameters for a hypoexponential variable with complex time distribution without parents as shown in Fig. 6.8a and in the presence of one exponential parent as shown in Fig. 6.8b. The underlying time distribution was approximated by using the proposed HCTBNs in this chapter and the standard CTBNs. The dynamics of the hypoexponential variable X in the learned models

1 http://rlair.cs.ucr.edu/ctbnrle/

2 http://home.math.au.dk/asmus/pspapers.html



(a)



(b)

Fig. 6.10: Probability of X staying at state 1, given evidence $X = 2$ and $Y = 2$ at time 8, 10 and 12. (6.10a): the true process has 10-order hypoexponential distribution and no parents. (6.10b): the true process has 5-order hypoexponential distribution and one parent. The rates in the distribution follow a *Gamma distribution* with rate = 1 and shape = 2. The number of hidden states for the learned HCTBNs is indicated by the number n.

and the underlying models are shown in Fig. 6.10. The results suggest that HCTBNs have a better approximation of the underlying hypoexponential distribution than CTBNs. It also indicates that other complex distributions may be better approximated using HCTBNs. In addition, we obtained a better approximation by increasing the number of hidden states. More importantly, the memory in the underlying temporal processes can be easily captured by HCTBNs, but not by CTBNs.

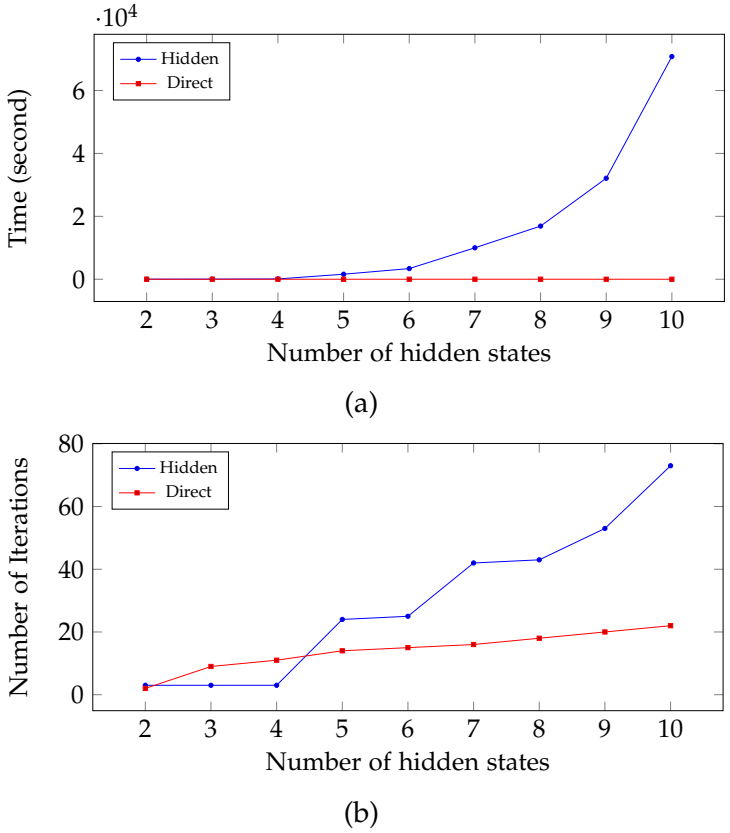


Fig. 6.11: Learning for COPD networks using the direct and hidden approaches with the number of hidden states ranging from 2 to 10 in the learned models, while is fixed to 4 in the underlying model: the time until convergence in (a) and the number of iterations until convergence in (b).

For the second part of the experiments, we considered learning parameters for a more complicated and realistic HCTBN for the COPD network as given in Fig. 6.4, where we have six variables, two of which are hypoexponential variables. The synthetic training datasets for the COPD network consisted of approximately 6000 observations on aver-

age. The number of hidden states for hidden variables H_1 and H_2 in the underlying model was both set to 4.

The number of iterations and the time until the EM algorithm converges are shown in Fig. 6.11a and Fig. 6.11b, respectively. The results in Fig. 6.11a suggest that the time until the EM algorithm converges grows exponentially with the number of states of the hidden variable using existing CTBNs learning algorithms, whereas there is little impact of the choice of the hidden states on the equivalent direct models (see the exponentially increasing time for HCTBNs when the number of hidden states increases from 6). Similarly, the number of iterations until convergence grows faster using existing CTBN learning algorithms than the equivalent direct models, whereas the difference is relatively small. By combining the results in Fig. 6.11a and Fig. 6.11b together, we can conclude that the time for each iteration in CTBNs grows exponentially with the number of hidden states. This is mainly attributed to the relatively large number of variables in the models, leading to a significant increase in the computation at each iteration.

We further evaluated the quality of the learned models in terms of log-likelihood. To rule out the randomness of starting parameters in the EM algorithms, we learned models with 30 different starting parameters for each approach. To test the performance for general models, these learning methods were used to estimate parameters for 30 underlying models with the same structure but different parameters. The test data were generated independently from the training data and the log-likelihood difference was subtracted the mean log-likelihood on the test data of learned models from those of the underlying models, as shown in Fig. 6.12. A lower log-likelihood difference indicates higher quality of the learned models. It is clear that the hidden approach has a better and more stable estimation of parameters for the underlying models. We also obtained the $p-$value < 0.05 based on a paired $T-$test for the mean log-likelihood for both methods, which suggests that the quality of the learned models achieved by the hidden approach is significantly higher than the direct approach.

6.7 CONCLUSIONS

In this paper, we provide a new formalization HCTBN where time duration is governed by hypoexponential distributions by employing auxiliary hidden variables. We also demonstrate that these hidden variables

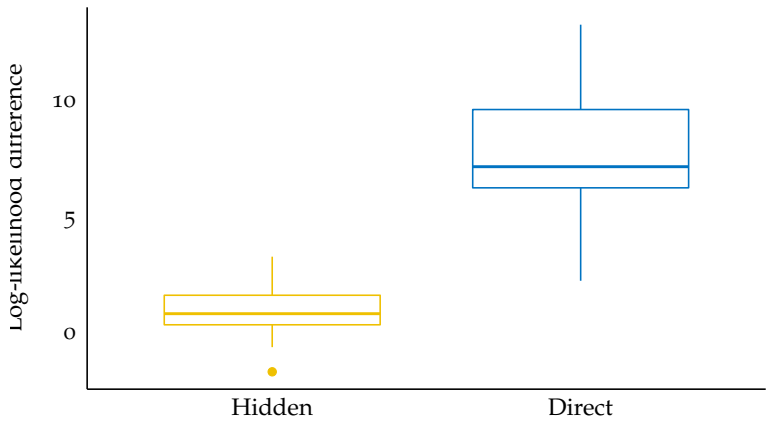


Fig. 6.12: Test log-likelihood for learned models using the direct and hidden approaches from 30 different underlying models. For a given underlying model, models are learned using each approach with 30 different starting parameters for the EM algorithms. The number of hidden states in both learned models and underlying models is set to 4. The log-likelihood difference is computed by subtracting the mean log-likelihood of learned models from those of the underlying models. Lower values indicate a better parameter estimation. The p-value < 0.05 was computed using the paired t-test on the mean log-likelihood of the learned models using direct and hidden approaches.

introduce *memory*, which is lacking in standard CTBNs. This memory will make CTBNs better-suited as a modeling tool for more general real-world problems in many domains, such as biology where memory plays a central role. In addition, the experimental results show that HCTBNs indeed can learn these more complex distributions. Transforming the learning task into equivalent direct models has the advantage of lower computational cost at the expense of the quality of learned models. Nevertheless, such an approach is infeasible when some observable variables, i.e., the hypoexponential and exponential variables, are partially known.

A limitation of HCTBNs so far is that the hypoexponential variables are restricted to be two-valued, as the focus of this paper has been on introducing a richer time distribution and memory. In future work, we aim to overcome this limitation by introducing more states for the hidden variables to allow hypoexponential variables to transition from one state to any other state. Using existing learning algorithms in CTBNs has the advantage of learning from partial trajectories, where observable variables are not fully known. However, these algorithms so far suffer from costly computation time. One possible solution to solve this problem is to decompose the EM algorithm into two parts, one part consisting of partially observed variables another part consisting of fully observed variables. This decomposition will significantly reduce the computation time, in particular when the exponential variables and their parents are fully observed.

7

CONCLUSIONS AND FURTHER RESEARCH

In this thesis, we have investigated various methods to model complex dynamic systems that evolve over time. Most of these methods are based on continuous time Bayesian networks (CTBNs), which are powerful probabilistic tools to model complex dynamic systems. In this chapter we take a step back, and summarize what we see as the main contributions of the research in the field of CTBNs. In addition, we provide some views on possible directions for future research topics.

7.1 MAIN CONTRIBUTIONS

CTBNs are an elegant modeling language for complex dynamic systems that evolve over continuous time. CTBNs are suitable for querying a probability distribution at arbitrary time points when particular events of interest occur. In addition, dynamic systems that consist of parts that evolve at different rates can be modeled. Nevertheless, modeling time as a continuous parameter is not always appropriate, in particular when only expert knowledge in a domain is available. In this case, it is a hurdle to derive parameters from expert knowledge to construct CTBNs. Besides, the exponential time duration distribution, as it is assumed in CTBNs, is not always appropriate. To address these issues, we have developed two useful extensions of the existing CTBN formalism to better deal with real-world problems that require more general time durations.

One extension is a hybrid model of DBNs and CTBNs, where time can be both discrete and continuous in the model. This is a significant contribution as each part of dynamic systems can be modeled using its own appropriate mechanism. Parts of systems that change regularly can be modeled as DBNs, and thus expert domain knowledge can be used to facilitate the modeling process. Meanwhile, other parts of the system

that transition at continuous rates can be represented as CTBNs, where such rates are well captured.

We also relax the assumption that the time duration follows an exponential distribution in CTBNs. Instead, we extend CTBNs to support a richer and more flexible time duration, as described in Chapter 6. The research in this thesis demonstrates that the proposed models can indeed better capture more complex underlying distributions that can not be captured by the exponential distribution. Furthermore, another significant contribution is that the proposed models also introduce memory behavior, which is lacking in the existing CTBNs. This is an important feature for CTBNs to be more competitive as a modeling tool as it allows modelling dynamic system where the future behaviour is determined by events from the past.

Theoretically, these two extensions make the existing CTBNs more powerful and better suited to a wider spectrum of real-world problems. Practically, we also have investigated the use of DBNs and CTBNs to model a medical problem from real data. It was shown that CTBNs are at least competitive with DBNs in terms of modeling both regular and irregularly spaced observations. Moreover, CTBNs provide more detailed temporal information.

7.2 FURTHER RESEARCH

The work described in this thesis leaves various opportunities for further research. In this section, we will briefly discuss some of them.

For hybrid time Bayesian networks, supporting both regularly and irregularly changing parts of systems, there is still a need to develop more efficient inference algorithms. Currently, the only inference approach is to transform these models into an equivalent Bayesian network. On the one hand, this approach is indeed an advantage as the existing inference algorithms in Bayesian networks can be directly applied. On the other hand, it can come at a high computational cost. For this reason we tried to avoid the use of the approach in the EM algorithm, where the transformation is needed at each iteration. Thus, one of interesting future research questions is to seek more efficient inference algorithms.

For the second extension, hypoexponential CTBNs, variables with a more complex time distribution are currently restricted to be binary. This restriction is mainly due to the fact that two or more variables in CTBNs are not allowed to change at the same time. Another interest-

ing research direction involves a comparison to neural networks that model time series, which are currently widely used. As graphical models, CTBNs provide a clear representation of the structure of a problem of interest, whereas neural networks act as a black box in the sense that they give us little insight in the underlying structure and meaning of the model, which limits interpretation and explanation of the results.

Another possible research question for the second proposed models is whether the existing EM algorithm can be further decomposed. It has been shown in this thesis that the existing EM algorithm in CTBNs is rather computationally expensive at each iteration. In the current EM algorithm, the expectation has to be computed for all variables in the model, even for those which are themselves fully observed and their parent variables as well. One possible solution is to decompose the EM algorithm by learning parameters based on sufficient statistics, rather than using iteratively computed expectations. In that case the expectation of fully observed variables does not change in the entire learning procedure.

A

EVIDENCE TYPES AND INFERENCE IN CTBNS

The evidence type in CTBNs, i.e. point and interval evidence, is a relevant factor for inference algorithms to answer a query over time. In the following, we first give two inference algorithms corresponding to point and interval evidence in CTBNs. It is then followed by a concrete example to illustrate the difference between these two types of evidence.

Consider a CTBN over a set of random variables $\mathbf{X}$ with a joint intensity matrix $Q_{\mathbf{X}}$. We are interested in querying the probability distribution of variables $\mathbf{X}$ at time t given evidence $\mathbf{Z} = \mathbf{z}$ before time t, $\mathbf{Z} \subseteq \mathbf{X}$. To query the distribution at a particular time point t as given in Equation 2.7, we have

$$P(\mathbf{X}_t \mid \mathbf{Z}_s = \mathbf{z}) = \exp(Q_{\mathbf{X}}(t-s)), \quad s < t \tag{A.1}$$

where $Q_{\mathbf{X}}$ is the joint intensity matrix over variables $\mathbf{X}$, a mixture of conditional intensity matrices for variables $\mathbf{X}$ incorporated with the evidence $\mathbf{Z} = \mathbf{z}$.

When the evidence $\mathbf{Z} = \mathbf{z}$ is interpreted as point evidence, i.e., we only know that variables $\mathbf{Z}$ have the values of $\mathbf{z}$ at a particular time point, such as at time s, $s < t$, and the values of $\mathbf{Z}$ in the open time interval (s, t) are unknown. This also implies that the transitions of variables $\mathbf{Z}$ in the interval $[s, t)$ can start at an arbitrary time and end at an arbitrary time. Besides, variables $\mathbf{Z}$ can have zero or more than one transitions from one state to another in the given interval $[s, t)$. For point evidence, the intensity matrix $Q_{\mathbf{X}}$ in Equation A.1 is the joint intensity matrix $Q^p_{\mathbf{X}}$ computed by the amalgamation over all conditional intensity matrices in the CTBN.

The situation is different when the evidence before time t is interpreted as interval evidence, i.e., variables $\mathbf{Z}$ have the values $\mathbf{z}$ in the half-open interval $[s, t)$. In this situation, the dynamics of the system are governed by a subset of conditional intensity matrices in the model, rather than intensity matrices for all the variables. Alternatively, it can

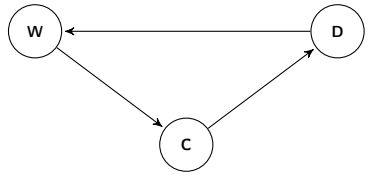


Fig. A.1: A CTBN over three variables W, C, D.

be viewed that the transitions in the time interval $[s, t)$ that are inconsistent with evidence $\mathbf{Z} = \mathbf{z}$ are blocked. In other words, the transitions for variables $\mathbf{Z}$ between state $\mathbf{z}$ and $\mathbf{z}'$, $\mathbf{z} \neq \mathbf{z}'$, are not allowed in the interval $[s, t)$.

To account for the interval evidence, it is addressed by zeroing out non-diagonal intensities in $Q^p_{\mathbf{X}}$ that correspond to impossible transitions for variables $\mathbf{Z}$. This approach was firstly discussed in [70] where expectation propagation was proposed to reason using interval evidence. Incorporated with evidence $\mathbf{Z} = \mathbf{z}$, we get a new joint intensity matrix $Q^i_{\mathbf{X}}$, $Q^i_{\mathbf{X}} \neq Q^p_{\mathbf{X}}$. Note that in most cases each row $Q^i_{\mathbf{X}}$ does not sum up to 0. Instead, it sums up to negative numbers. This gives us an unnormalized distribution characterized by $Q^i_{\mathbf{X}}$.

In the following, we further illustrate the inference difference using point and interval evidence with a concrete example.

Example A.1. *Consider a CTBN over three variables W, C, D with its graph structure as shown in Fig. A.1. The intensity matrices for the CTBN are given in Fig. A.2. Variables change quickly in some situations and change slowly in other situations, which can be dependent on their parents. For example, the intensity matrices for C, $Q_{C|W=w}$ and $Q_{C|W=\overline{w}}$ indicate that, the average time for variable C transitioning from state c to $\overline{c}$ when $W = w$ is ten times (0.001/0.0001) as its transition from state $\overline{c}$ to c when $W = \overline{w}$.*

Given the variables W, C, D, the variable order $W < C < D$ and the order on their states $\overline{w} < w$, $\overline{c} < c$, $\overline{d} < d$, we have a sequence of states over variables W, C, D in the following, which is also the order of their joint intensity matrix:

$$\langle \overline{w}, \overline{c}, \overline{d} \rangle, \langle w, \overline{c}, \overline{d} \rangle, \langle \overline{w}, c, \overline{d} \rangle, \langle w, c, \overline{d} \rangle, \langle \overline{w}, \overline{c}, d \rangle \langle w, \overline{c}, d \rangle, \langle \overline{w}, c, d \rangle, \langle w, c, d \rangle$$

We are also given a uniform distribution as prior distribution, which will be later used to compute the joint distribution:

$$P(W_0, C_0, D_0) = (0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125)$$

$$Q_{\mathrm{W|D}=\overline{d}} = \begin{array}{c} \\ \overline{w} \\ w \end{array} \begin{array}{c} \begin{array}{cc} \overline{w} & w \end{array} \\ \begin{pmatrix} -0.17 & 0.17 \\ 100 & -100 \end{pmatrix} \end{array} \qquad Q_{\mathrm{W|D}=d} = \begin{array}{c} \\ \overline{w} \\ w \end{array} \begin{array}{c} \begin{array}{cc} \overline{w} & w \end{array} \\ \begin{pmatrix} -2 & 2 \\ 18 & -18 \end{pmatrix} \end{array}$$

$$Q_{\mathrm{D|C}=\overline{c}} = \begin{array}{c} \\ \overline{d} \\ d \end{array} \begin{array}{c} \begin{array}{cc} \overline{d} & d \end{array} \\ \begin{pmatrix} -0.8 & 0.8 \\ 0.001 & -0.001 \end{pmatrix} \end{array} \qquad Q_{\mathrm{D|C}=c} = \begin{array}{c} \\ \overline{d} \\ d \end{array} \begin{array}{c} \begin{array}{cc} \overline{d} & d \end{array} \\ \begin{pmatrix} -0.01 & 0.01 \\ 5 & -5 \end{pmatrix} \end{array}$$

$$Q_{\mathrm{C|W}=\overline{w}} = \begin{array}{c} \\ \overline{c} \\ c \end{array} \begin{array}{c} \begin{array}{cc} \overline{c} & c \end{array} \\ \begin{pmatrix} -0.001 & 0.001 \\ 0.5 & -0.5 \end{pmatrix} \end{array} \qquad Q_{\mathrm{C|W}=w} = \begin{array}{c} \\ \overline{c} \\ c \end{array} \begin{array}{c} \begin{array}{cc} \overline{c} & \mathrm{c} \end{array} \\ \begin{pmatrix} -5 & 5 \\ 0.0001 & -0.0001 \end{pmatrix} \end{array}$$

Fig. A.2: Conditional intensity matrices for a CTBN over three variables W, C, D with its structure as given in Fig. A.1. The intensity matrices for W given D (top), D given C (middle) and C given W (bottom).

Example A.2. *We reconsider the Example A.1 to query the dynamics of variables using point evidence and interval evidence. Assume we have point evidence $D = d$ at time 6, 7, 8, $W = w$ at time 8, 10, 13. By comparison, we have interval evidence $D = d$ in the interval $[6, 8)$ and $W = w$ in the interval $[8, 13)$. We want to compute the distribution in the interval $[0, 20]$ using point and interval evidence.*

For illustrative purpose, we compute the distribution in the time interval $[6, 8)$ and the distribution in other intervals can be computed similarly. To compute probability distribution in the interval $[6, 8)$ using point and interval evidence, we first need to obtain their corresponding intensity matrices used to answer the query. For the point evidence $D = d$ at time 6, the joint intensity matrix is computed by the amalgamation on *all* the conditional intensity matrices given in Fig. A.2, resulting in the joint intensity matrix Q_{WCD} as shown in the following:

$$\begin{pmatrix}
-0.971 & 0.17 & 0.001 & 0 & 0.8 & 0 & 0 & 0 \\
100 & -105.8 & 0 & 5 & 0 & 0.8 & 0 & 0 \\
0.5 & 0 & -0.68 & 0.17 & 0 & 0 & 0.01 & 0 \\
0 & 0.0001 & 100 & -100.0101 & 0 & 0 & 0 & 0.01 \\
0.001 & 0 & 0 & 0 & -2.002 & 2 & 0.001 & 0 \\
0 & 0.001 & 0 & 0 & 18 & -23.001 & 0 & 5 \\
0 & 0 & 5 & 0 & 0.5 & 0 & -7.5 & 2 \\
0 & 0 & 0 & 5 & 0 & 0.0001 & 18 & -23.0001
\end{pmatrix}$$

For the interval evidence $D = d$ in the time interval $[6, 8)$, however, the dynamics of the system are governed by a new intensity matrix which is computed by zeroing out rows and columns in the matrix Q_{WCD} that are inconsistent with evidence $D = d$, i.e., we zero out the rows (from 1 to 4) and columns (from 1 to 4), resulting the following joint intensity matrix Q'_{WCD}, $Q'_{\text{WCD}} \neq Q_{\text{WCD}}$.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2.002 & 2 & 0.001 & 0 \\
0 & 0 & 0 & 0 & 18 & -23.001 & 0 & 5 \\
0 & 0 & 0 & 0 & 0.5 & 0 & -7.5 & 2 \\
0 & 0 & 0 & 0 & 0 & 0.0001 & 18 & -23.0001
\end{pmatrix}$$

Obviously, the distributions for variables W, C, D after time 6 given point evidence $D = d$ at time 6 and interval evidence $D = d$ in the

interval $[6, 8)$, as shown in Fig. A.3, are different as the system dynamics are determined by two different intensity matrices Q_{WCD} and Q'_{WCD}, $Q_{\text{WCD}} \neq Q'_{\text{WCD}}$. Another difference can also be seen in the time interval $[8, 13)$ where we have point evidence $W = w$ at time 8, 10 and interval evidence $W = w$ in the interval $[8, 13)$.

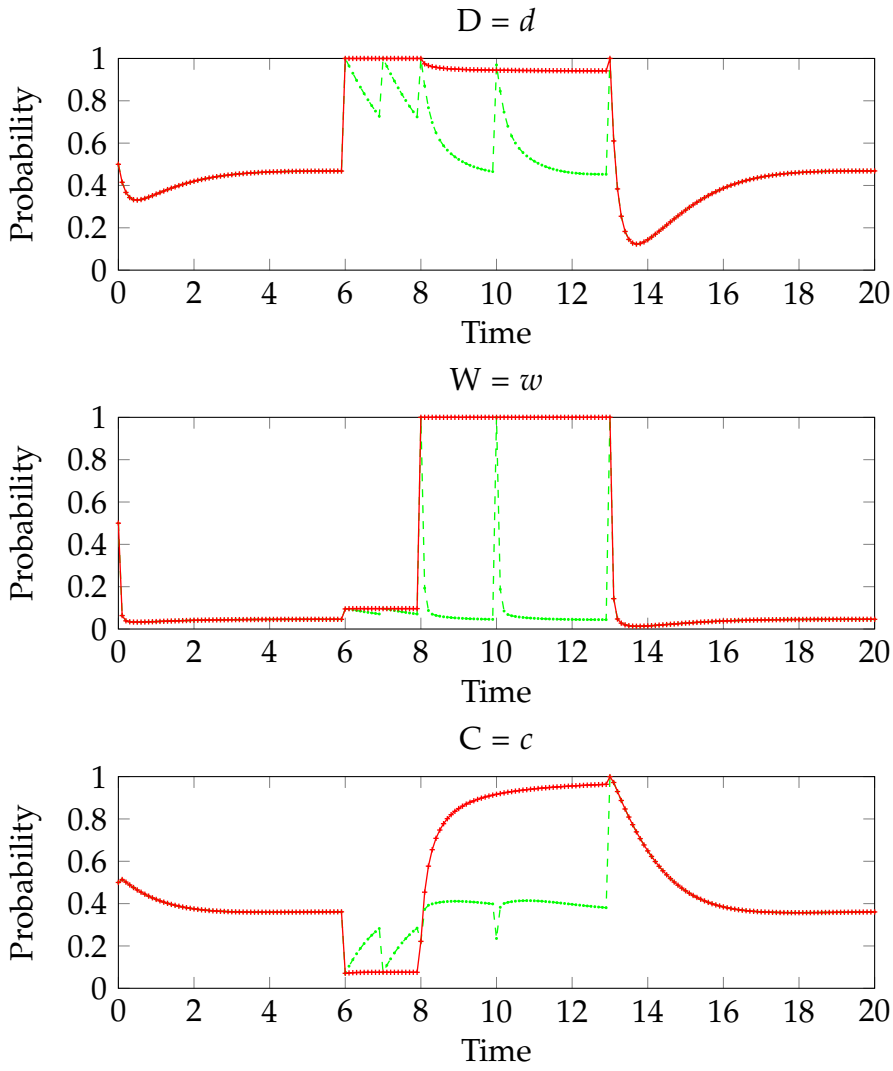


Fig. A.3: Marginal distributions for variables in a CTBN over theree variables W, C, D in the time interval $[0.20]$ using point and interval evidence. In the interval $[0, 13)$, we have point evidence) D = d at time 6, 7, 8 and W = w at time 8,10, and interval evidence) D = d in the interval [6,8) and W = w in the interval $[8, 13)$. In the interval $[13, 20]$, we only point evidence only at time 13 for all variables in the model, i.e., the values of W, C, D are known at time 13. In this case, we have W = w, D = d, C = c at time 13. Green line indicates the distribution given point evidence and red given interval evidence.

BIBLIOGRAPHY

[1] Enzo Acerbi, Teresa Zelante, Vipin Narang, and Fabio Stella. Gene network inference using continuous time Bayesian networks: a comparative study and application to Th17 cell differentiation. *BMC Bioinformatics* 15.1 (2014), pp. 1–27 (cit. on pp. 84, 85).

[2] Enzo Acerbi, Elena Viganò, Michael Poidinger, Alessandra Mortellaro, Teresa Zelante, and Fabio Stella. Continuous time Bayesian networks identify Prdm1 as a negative regulator of TH17 cell differentiation in humans. *Scientific reports* 6 (2016) (cit. on p. 84).

[3] Klaus-Peter Adlassnig, Carlo Combi, Amar K. Das, Elpida T. Keravnou, and Giuseppe Pozz. Temporal representation and reasoning in medicine: Research directions and challenges. *Artificial Intelligence in Medicine* 38 (2006), pp. 101–113 (cit. on p. 85).

[4] N.R Anthonisen, J. Manfreda, C. P. W. Warren, E. S. Hershfield, G. K. M. Harding, and N. A. Nelson. Antibiotic therapy in exacerbations of chronic obstructive pulmonary disease. *Annals of internal medicine* 106.2 (1987), pp. 196–204 (cit. on p. 83).

[5] Norman T. J. Bailey. A statistical method of estimating the periods of incubation and infection of an infectious disease. *Nature* 174.4420 (1954), p. 139 (cit. on pp. 116, 117).

[6] Jim Basilakis, Nigel H. Lovell, Stephen J. Redmond, and Branko G. Celler. Design of a Decision-Support Architecture for Management of Remotely Monitored Patients. *IEEE Transactions on Information Technology in Biomedicine* 14.5 (2010), pp. 1216–1226 (cit. on p. 83).

[7] Claudio Bettini, Sushil Jajodia, and Sean Wang. *Time granularities in databases, data mining, and temporal reasoning*. Springer, 2000 (cit. on p. 32).

[8] Erik W.M.A. Bischoff, Lonneke M. Boer, Johan Molema, Reinier Akkermans, Chris van Weel, Jan H. Vercoulen, and Tjard R.J. Schermer. Validity of an automated telephonic system to assess COPD exacerbation rates. *European Respiratory Journal* 39.5 (2012), pp. 1090–1096 (cit. on pp. 80, 81, 83).

[9] Mogens Bladt. A review on phase-type distributions and their use in risk theory. *ASTIN Bulletin* 35.1 (2005), pp. 145–161 (cit. on p. 119).

[10] Hichem Boudali and Joanne Bechta Dugan. A continuous-time Bayesian network reliability modeling, and analysis framework. *IEEE Transactions on Reliability* 55.1 (2006), pp. 86–97 (cit. on p. 84).

[11] Carlo Combi and Yuval Shahar. Temporal reasoning and temporal data maintenance in medicine: Issues and challenges. *Computers in Biology and Medicine* 27.5 (1997). Time-oriented Systems in Medicine, pp. 353 –368 (cit. on p. 85).

[12] Gregory E. Cooperhic and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from databases. *Machine Learning* 9 (1992), pp. 309–347 (cit. on p. 9).

[13] James H. Martin Daniel Jurafsky. *Hidden Markov Models*. 2016 (cit. on p. 61).

[14] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence* 5.2 (1989), pp. 142–150 (cit. on pp. 9, 23, 55).

[15] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38 (cit. on pp. 47, 60).

[16] F.T. de Dombal, D.J. Leaper, J.R. Staniland, A.P. McAnn, and J.C. Horrocks. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal* (1972), pp. 9–13 (cit. on p. 4).

[17] Joanna Dunkley, Martin Bucher, Pedro G Ferreira, Kavilan Moodley, and Constantinos Skordis. Fast and reliable Markov chain Monte Carlo technique for cosmological parameter estimation. *Monthly Notices of the Royal Astronomical Society* 356.3 (2005), pp. 925–936 (cit. on p. 62).

[18] Daniel Eaton and Kevin Murphy. Bayesian structure learning using dynamic programming and MCMC. *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. UAI'07. Vancouver, BC, Canada: AUAI Press, 2007, pp. 101–108 (cit. on p. 62).

[19] Andreas Eckner. A framework for the analysis of unevenly spaced time series data. *Preprint. Available at:* `http://www.eckner.com/papers/unevenly_spaced_time_series_analysis` (2012) (cit. on p. 93).

[20] Tanja W Effing, Huib AM Kerstjens, Evelyn M Monninkhof, Paul DLPM van der Valk, Emiel FM Wouters, Dirkje S Postma, Gerhard A Zielhuis, and Job van der Palen. Definitions of exacerbations: does it really matter in clinical trials on COPD? *CHEST Journal* 136.3 (2009), pp. 918–923 (cit. on pp. 81, 83).

[21] Mark A. Espeland, Orah S. Platt, and Dianne Gallagher. Joint estimation of incidence and diagnostic error rates from irregular longitudinal data. *Journal of the American Statistical Association* 84.408 (1989), pp. 972–979 (cit. on p. 85).

[22] Yu Fan and Christian R. Shelton. Learning Continuous-time Social Network Dynamics. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI '09. Montreal, Quebec, Canada: AUAI Press, 2009, pp. 161–168 (cit. on p. 55).

[23] Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. *Proceedings of the Fourteenth International Conference on Machine Learning*. Vol. 97. 1997, pp. 125–133 (cit. on p. 61).

[24] Nir Friedman. The Bayesian structural EM algorithm. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 129–138 (cit. on p. 61).

[25] Nir Friedman and Daphne Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* 50.1 (2003), pp. 95–125 (cit. on p. 62).

[26] E. Gatti, D. Luciani, and F. Stella. A continuous time Bayesian network model for cardiogenic heart failure. *Flexible Services and Manufacturing Journal* 24.4 (2012), pp. 496–515 (cit. on pp. 55, 58, 84).

[27] Walter R Gilks. *Markov chain Monte Carlo*. Wiley Online Library, 2005 (cit. on p. 61).

[28] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101.23 (2000), e215–e220 (cit. on p. 50).

[29] Karthik Gopalratnam, Henry Kautz, and Daniel S Weld. Extending continuous time Bayesian networks. *Proceedings of the national conference on artificial intelligence*. Vol. 20. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2005, p. 981 (cit. on pp. 85, 114, 132).

[30] G. Anthony Gorry and G. Octo Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research* 1 (1968), pp. 490–507 (cit. on p. 4).

[31] K. J. Gough. The estimation of latent and infectious periods. *Biometrika* 64.3 (1977), pp. 559–565 (cit. on pp. 116, 117).

[32] Maria Adela Grando, Ronen Rozenblum, and David Bates, eds. *Information Technology for Patient Empowerment in Healthcare*. Walter de Gruyter, Berlin/Boston/Munich, 2015 (cit. on p. 80).

[33] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes, 3rd edition*. Oxford University Press, 2001 (cit. on p. 14).

[34] Marco Grzegorczyk and Dirk Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning* 71.2 (2008), p. 265 (cit. on p. 62).

[35] Marco Grzegorczyk and Dirk Husmeier. Non-stationary continuous dynamic Bayesian networks. *Advances in Neural Information Processing Systems*. 2009, pp. 682–690 (cit. on p. 55).

[36] Olle Häggström. *Finite Markov chains and algorithmic applications*. Vol. 52. Cambridge University Press, 2002 (cit. on p. 20).

[37] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57.1 (1970), pp. 97–109 (cit. on p. 61).

[38] David E. Heckerman, Eric J. Horvitz, and Bharat N. Nathwani. Towards normative expert systems: part I – The Pathfinder project. *Methods of Information in Medicine* 31 (1992), pp. 90–105 (cit. on p. 9).

[39] David E. Heckerman and Bharat N. Nathwani. Towards normative expert systems: part II – probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine* 31 (1992), pp. 106–116 (cit. on p. 9).

[40] Maarten van der Heijden and Arjen Hommersom. Causal independence models for continuous time Bayesian networks. *Probabilistic Graphical Models: 7th European Workshop, PGM 2014, Utrecht, The Netherlands, September 17-19, 2014. Proceedings*. Ed. by Linda C. van der Gaag and Ad J. Feelders. Cham: Springer International Publishing, 2014, pp. 503–518 (cit. on p. 56).

[41] Maarten van der Heijden, Marina Velikova, and Peter J. F. Lucas. Learning Bayesian networks for clinical time series analysis. *Journal of Biomedical Informatics* 48 (2014), pp. 94 –105 (cit. on pp. 80, 83, 84, 88, 98).

[42] Maarten van der Heijden, Peter J. F. Lucas, Bas Lijnse, Yvonne F. Heijdra, and Tjard R. J. Schermer. An autonomous mobile system for the management of COPD. *Journal of Biomedical Informatics* 46.3 (2013), pp. 458–469 (cit. on pp. 7, 18, 80, 83, 109).

[43] Arjen Hommersom and Peter J. F. Lucas. *Foundations of Biomedical Knowledge Representation: methods and applications*. Springer, 2015 (cit. on p. 5).

[44] John R. Hurst, Gavin C. Donaldson, Jennifer K. Quint, James JP Goldring, Anant RC Patel, and Jadwiga A. Wedzicha. Domiciliary pulse-oximetry at exacerbation of chronic obstructive pulmonary disease: prospective pilot study. *BMC Pulmonary Medicine* 10.1 (2010), pp. 1–11 (cit. on pp. 80, 83, 86).

[45] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003 (cit. on p. 14).

[46] Morten H. Jensen, Simon L. Cichosz, Birthe Dinesen, and Ole K. Hejlesen. Moving prediction of exacerbation in chronic obstructive pulmonary disease for patients in telecare. eng. *Journal of Telemedicine and Telecare* 18 (2 2012), pp. 99–103 (cit. on pp. 83, 84).

[47] Elpida T. Keravnou. Medical temporal reasoning (editorial). *Artificial Intelligence in Medicine* 3.6 (1991), pp. 289–290 (cit. on p. 85).

[48] Elpida T. Keravnou. Medical temporal reasoning (editorial). *Artificial Intelligence in Medicine* 8.3 (1996), pp. 187–191 (cit. on p. 85).

[49] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009 (cit. on pp. 17, 61).

[50] James A de Lemos, Mark H Drazner, Torbjorn Omland, Colby R Ayers, Amit Khera, Anand Rohatgi, Ibrahim Hashim, Jarett D Berry, Sandeep R Das, David A Morrow, et al. Association of troponin T detected with a highly sensitive assay and cardiac structure and mortality risk in the general population. *JAMA* 304.22 (2010), pp. 2503–2512 (cit. on p. 51).

[51] David A Levin and Yuval Peres. *Markov chains and mixing times*. American Mathematical Soc., 2017 (cit. on p. 20).

[52] Haiqun Lin, Daniel O. Scharfstein, and Robert A. Rosenheck. Analysis of longitudinal data with irregular, outcome-dependent follow-up. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66.3 (2004), pp. 791–813 (cit. on p. 85).

[53] Manxia Liu, Arjen Hommersom, Maarten van der Heijden, and Peter J. F. Lucas. Hybrid time Bayesian networks. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. Ed. by Sébastien Destercke and Thierry Denoeux. Cham: Springer International Publishing, 2015, pp. 376–386 (cit. on p. 11).

[54] Manxia Liu, Arjen Hommersom, Maarten van der Heijden, and Peter J.F. Lucas. Learning parameters of hybrid time Bayesian networks. *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*. Ed. by Alessandro Antonucci, Giorgio Corani, and Cassio Polpo Campos. 2016, pp. 287–298 (cit. on p. 11).

[55] Manxia Liu, Arjen Hommersom, Maarten van der Heijden, and Peter J.F. Lucas. Hybrid time Bayesian networks. *Int. J. Approx. Reasoning* 80.C (Jan. 2017), pp. 460–474 (cit. on pp. 11, 61, 84).

[56] Manxia Liu, Fabio Stella, Arjen Hommersom, and Peter J. F. Lucas. Representing Hypoexponential Distributions in Continuous Time Bayesian Networks. *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*. Ed. by Jesús Medina, Manuel Ojeda-Aciego, José Luis Verdegay, Irina Perfilieva, Bernadette Bouchon-Meunier, and Ronald R. Yager. Cham: Springer International Publishing, 2018, pp. 565–577 (cit. on p. 12).

[57] Manxia Liu, Fabio Stella, Arjen Hommersom, and Peter J. F. Lucas. Making continuous time Bayesian networks more flexible. *Proceedings of the Nineth International Conference on Probabilistic Graphical Models*. in press (cit. on p. 12).

[58] Manxia Liu, Fabio Stella, Arjen Hommersom, Peter J. F. Lucas, Lonneke Boer, and Erik Bischoff. Modeling clinical time series data using temporal Bayesian networks. *Artificial Intelligence in Medicine* (under review) (cit. on p. 11).

[59] Peter J. F. Lucas and Linda C. van der Gaag. *Principles of Expert Systems*. Addison-Wesley, Wokingham, 1991 (cit. on p. 4).

[60] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. *International Conference on Machine Learning*. 2015, pp. 2113–2122 (cit. on p. 96).

[61] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007 (cit. on p. 60).

[62] Merle H Mishel. The measurement of uncertainty in illness. *Nursing research* (1981) (cit. on p. 2).

[63] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on p. 48).

[64] Kevin Patrick Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis. 2002 (cit. on pp. 9, 23, 32, 55, 62).

[65] Radhakrishnan Nagarajan, Marco Scutari, and Sophie Lèbre. *Bayesian Networks in R: With Applications in Systems Biology*. Springer Publishing Company, Incorporated, 2013 (cit. on p. 9).

[66] Richard E Neapolitan. *Probabilistic reasoning in expert systems: theory and algorithms*. CreateSpace Independent Publishing Platform, 2012 (cit. on p. 17).

[67] Kevin Murphy Nir Friedman and Stuart Russell. Learning the structure of dynamic probabilistic networks. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 139–147 (cit. on p. 61).

[68] Uri Nodelman. Continuous time Bayesian networks. PhD thesis. Stanford University, 2007 (cit. on pp. 9, 26).

[69] Uri Nodelman and Eric Horvitz. Continuous time Bayesian networks for inferring users' presence and activities with extensions for modeling and evaluation. *Microsoft Research, July-August* (2003) (cit. on p. 114).

[70] Uri Nodelman, Daphne Koller, and Christian R. Shelton. Expectation propagation for continuous time Bayesian networks. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. UAI'05. Edinburgh, Scotland: AUAI Press, 2005, pp. 431–440 (cit. on p. 144).

[71] Uri Nodelman, Christian R Shelton, and Daphne Koller. Continuous time Bayesian networks. *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. 2002, pp. 378–387 (cit. on pp. 23, 32, 43, 55).

[72] Uri Nodelman, Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*. UAI'03. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003, pp. 451–458 (cit. on p. 62).

[73] Uri Nodelman, Christian R. Shelton, and Daphne Kollerthu. Expectation maximization and complex duration distributions for continuous time Bayesian networks. *Proceedings of the Twenty-First International Conference on Uncertainty in Artificial Intelligence*. 2005, pp. 421–430 (cit. on pp. 56, 61, 114, 115).

[74] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988 (cit. on pp. 2, 17, 18, 32).

[75] Michael Ramati and Yuval Shahar. Irregular-time Bayesian networks. *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. UAI'10. Catalina Island, CA: AUAI Press, 2010, pp. 484–491 (cit. on pp. 56, 85).

[76] Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and extensions. *J. Mach. Learn. Res.* 14.1 (Jan. 2013), pp. 3295–3320 (cit. on p. 77).

[77] Kira Rehfeld, Norbert Marwan, Jobst Heitzig, and Jürgen Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics* 18.3 (2011), pp. 389–404 (cit. on p. 93).

[78] Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *Annals of Statistics* (2002), pp. 962–1030 (cit. on p. 43).

[79] Carsten Riggelsen. MCMC learning of Bayesian network models by Markov blanket decomposition. *Lecture notes in computer science* 3720 (2005), p. 329 (cit. on p. 62).

[80] Joshua W Robinson and Alexander J Hartemink. Non-stationary dynamic Bayesian networks. *Advances in Neural Information Processing Systems*. 2009, pp. 1369–1376 (cit. on p. 55).

[81] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care II (MIMIC-II): a public-access intensive care unit database. *Critical care medicine* 39.5 (2011), p. 952 (cit. on p. 50).

[82] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229 (cit. on p. 5).

[83] Daniel Sanchez-Morillo, Miguel Angel Fernandez-Granero, and Antonio León Jiménez. Detecting COPD exacerbations early using daily telemonitoring of symptoms and k-means clustering: a pilot study. *Medical and Biological Engineering and Computing* 53.5 (2015), pp. 441–451 (cit. on pp. 83, 84).

[84] Elizabeth Sapey and Robert A. Stockley. COPD exacerbations . 2: aetiology. *Thorax* 61.3 (2006), pp. 250–258 (cit. on p. 83).

[85] G. Schreiber, H. Akkermans, A. Anjewierden, R. De Hoog, N. R. Shadbolt, W. Van de Velde, and B. J. Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, Cambridge, MA, 1999 (cit. on p. 5).

[86] Terence AR Seemungal, Gavin C Donaldson, Angshu Bhowmik, Donald J Jeffries, and Jadwiga A Wedzicha. Time course and recovery of exacerbations in patients with chronic obstructive pulmonary disease. *American Journal of Respiratory and Critical Care Medicine* 161.5 (2000), pp. 1608–1613 (cit. on pp. 88, 117).

[87] Christian R. Shelton, Yu Fan, William Lam, Joon Lee, and Jing Xu. Continuous time Bayesian network reasoning and learning engine. *J. Mach. Learn. Res.* 11 (Mar. 2010), pp. 1137–1140 (cit. on p. 51).

[88] Michael Smithson. *Ignorance and uncertainty: emerging paradigms*. Springer Science & Business Media, 2012 (cit. on p. 2).

[89] David J. Spiegelhalter and Robin P. Knill-Jones. Statistical and knowledge-based approaches to clinical decision-support systems, with an application in gastroenterology. *Journal of the Royal Statistical Society. Series A (General)* 147.1 (1984), pp. 35–77 (cit. on p. 4).

[90] Arthur A Stone, Saul Shiffman, Joseph E Schwartz, Joan E Broderick, and Michael R Hufford. Patient compliance with paper and electronic diaries. *Controlled Clinical Trials* 24.2 (2003), pp. 182 –199 (cit. on p. 80).

[91] Gilbert Strang. *Linear algebra and its applications, 2nd edition*. Academic Press, New York, 1980 (cit. on p. 14).

[92] Liessman Sturlaugson and John W. Sheppard. Uncertain and negative evidence in continuous time Bayesian networks. *Int. J. Approx. Reasoning* 70.C (Mar. 2016), pp. 99–122 (cit. on p. 84).

[93] Chengwei Su and Mark E. Borsuk. Improving structure MCMC for Bayesian networks through Markov blanket resampling. *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 4042–4061 (cit. on p. 62).

[94] Z.M. Sund, T. Powell, R. Greenwood, and N.A. Jarad. Remote daily real-time monitoring in patients with COPD - A feasibility study using a novel device. *Respiratory Medicine* 103.9 (2009), pp. 1320 –1328 (cit. on p. 83).

[95] Annette ten Teije, Silvia Miksch, and Peter J. F. Lucas. *Computer-based medical guidelines and protocols: a primer and current trends*. 1st ed. Amsterdam, The Netherlands: IOS Press, 2008 (cit. on p. 5).

[96] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA: ACM, 2013, pp. 847–855 (cit. on p. 96).

[97] Marina Velikova, Peter J. F. Lucas, and Maarten van der Heijden. Intelligent disease self-management with mobile technology. *IEEE Computer* 48 (2015), pp. 32–39 (cit. on pp. 80, 109).

[98] Roel Verbelen. Phase-type distributions & mixtures of Erlangs. 2013 (cit. on pp. 118, 119).

[99] Simone Villa. *CTBN R interface*. 2014 (cit. on p. 51).

[100] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*. Macmillan New York, 1993 (cit. on p. 14).

[101] Jadwiga A. Wedzicha and Terence AR Seemungal. COPD exacerbations: defining their cause and prevention. *The Lancet* 370.9589 (2007), pp. 786–796 (cit. on p. 83).

[102] Jing Xu and Christian R. Shelton. Continuous time Bayesian networks for host level network intrusion detection. *Machine Learning and Knowledge Discovery in Databases*. Ed. by Walter Daelemans, Bart Goethals, and Katharina Morik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 613–627 (cit. on pp. 55, 84).

[103] Jing Xu and Christian R Shelton. Intrusion detection using continuous time Bayesian networks. *Journal of Artificial Intelligence Research* 39 (2010), pp. 745–774 (cit. on p. 84).

[104] Yu Zhang, Zhidong Deng, Hongshan Jiang, and Peifa Jia. Dynamic Bayesian Network (DBN) with structure expectation maximization (SEM) for modeling of gene network from time series gene expression data. *BIOCOMP*. 2006, pp. 41–47 (cit. on p. 98).

LIST OF SIKS DISSERTATIONS

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models

02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems

04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference

05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage

07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction

08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues

09 Tim de Jong (OU), Contextualised Mobile Media for Learning

10 Bart Bogaert (UvT), Cloud Content Contention

11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective

12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining

13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling

14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets

15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval

16 Maarten Schadd (UM), Selective Search in Games of Different Complexity

17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness

18 Mark Ponsen (UM), Strategic Decision-Making in complex games
19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles
20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29 Faisal Kamiran (TUE), Discrimination-aware Classification
30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33 Tom van der Weide (UU), Arguing to Motivate Decisions
34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training

36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach

37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization

39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games

40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development

41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control

42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution

43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge

44 Boris Reuderink (UT), Robust Brain-Computer Interfaces

45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection

46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression

48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

---

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda

02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories

04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications

05 Marijn Plomp (UU), Maturing Interorganisational Information Systems

06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks

07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories

09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms

10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment

11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems

13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems

15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.

16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance

18 Eltjo Poort (VU), Improving Solution Architecting Practices

19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution

20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval

22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41 Sebastian Kelle (OU), Game Design Patterns for Learning

42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
43 Withdrawn
44 Anna Tordai (VU), On Combining Alignment Techniques
45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

---

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support
02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
07 Giel van Lankveld (UvT), Quantifying Individual Player Differences
08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications

10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.

11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services

12 Marian Razavian (VU), Knowledge-driven Migration to Services

13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning

15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications

16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation

17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid

18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification

19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling

20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval

21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation

22 Tom Claassen (RUN), Causal Discovery and Logic

23 Patricio de Alencar Silva (UvT), Value Activity Monitoring

24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning

25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning

27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance

28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience

29 Iwan de Kok (UT), Listening Heads

30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support

31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications

32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere

34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search

35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction

36 Than Lam Hoang (TUe), Pattern Mining in Data Streams

37 Dirk Börner (OUN), Ambient Learning Displays

38 Eelco den Heijer (VU), Autonomous Evolutionary Art

39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems

40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games

41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning

43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data

02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method

03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions

04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability

06 Damian Tamburri (VU), Supporting Networked Software Development

07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior

08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints

09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

10 Ivan Salvador Razo Zapata (VU), Service Value Networks

11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support

12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control

13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

14 Yangyang Shi (TUD), Language Models With Meta-information

15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria

17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations

19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support

20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link

21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized k-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education

41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for network data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning

11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains

30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31 Yakup Koç (TUD), On the robustness of Power Grids
32 Jerome Gard (UL), Corporate Venture Management in SMEs
33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
34 Victor de Graaf (UT), Gesocial Recommender Systems
35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

---

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations

33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime

02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications

39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets

43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement

46 Jan Schneider (OU), Sensor-based Learning Support

47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration

48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TUE), Multi-perspective Process Mining

03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process

06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems

07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems

08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems

09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Using behavioral context in process mining: Exploring the added value of computational models for increasing the use of renewable energy in the residential sector
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VU), Emergent relational schemas for RDF

SUMMARY

This doctoral thesis is concerned with the theoretical and practical aspects of continuous-time and discrete-time Bayesian networks, in particular their use to model dynamic systems. The six research papers included in the thesis offer contributions to the existing theory of continuous time Bayesian networks, in addition to experimental investigations into the expressive power of different temporal Bayesian networks.

One of the theoretical contributions are the new notion of hybrid-time Bayesian networks, a combination of dynamic Bayesian networks and continuous-time Bayesian networks. In hybrid-time models, time is allowed to be either discrete or continuous. Parts of dynamic systems can be modeled as discrete-time Bayesian networks, and thus expert domain knowledge can be used to facilitate the modeling process. Other parts can be modeled as a continuous-time Bayesian network, which are very suitable when learning from data where variables are observed irregularly in time.

Another theoretical contribution to standard continuous-time Bayesian networks is obtained by relaxing the assumption that the time duration, i.e., the amount of time that a variable stays in a state before it transients to another, follows an exponential distribution. In the extended model, the exponential distribution is replaced by a hypoexponential distribution (the exponential distribution is one of its special cases), a richer and more flexible distribution. The research in this thesis demonstrates that continuous-time Bayesian networks with hypoexponential distributions can indeed capture more complex underlying distributions than the exponential distribution is capable of. Furthermore, two methods for parameter estimation from complete data are proposed and experimentally compared.

The experimental research is devoted to an in-depth investigation of the practical use of dynamic Bayesian networks and continuous-time Bayesian networks to model dynamic systems. The research makes use of a medical problem with real-world data to investigate several aspects of Bayesian network representations that one may need to take into consideration. This study sheds some light on the practical requirements of using these two temporal models in practice.

SAMENVATTING

Dit proefschrift betreft theoretische en praktische aspecten van continue-tijd Bayesiaanse netwerken om dynamische systemen te beschrijven. De zes wetenschappelijke artikelen in dit proefschrift resulteerden in bijdragen aan de bestaande theorie van continue-tijd Bayesiaanse netwerken en experimenteel onderzoek naar de uitdrukkingskracht van verschillende soorten temporele Bayesiaanse netwerken.

Eén van de theoretische bijdragen betreft de introductie van hybride-tijd Bayesiaanse netwerken waarbij dynamische Bayesiaanse netwerken en continue-tijd Bayesiaanse netwerken worden gecombineerd. Tijd kan zowel discreet of continu zijn in de voorgestelde hybride-tijd-modellen. Dynamische systemen kunnen gedeeltelijk worden gemodelleerd als discrete-tijd Bayesiaanse netwerken, waardoor kennis van domeinexperts het modeleerproces kunnen faciliteren. Andere delen kunnen worden gerepresenteerd als een continue-tijd Bayesiaanse netwerk dat geschikt is om te leren uit gegevens waarbij variabelen onregelmatig worden geobserveerd.

Standaard continue-tijd Bayesiaanse netwerken nemen aan dat de verblijftijd van het proces in een bepaalde toestand een exponentiële distributie volgt. In dit proefschrift wordt deze aanname verzwakt door de exponentiële distributie te vervangen door een hypoexponentiële distributie, een rijkere en meer flexibele distributie waarvan de exponentiële distributie één van de speciale gevallen is. Het onderzoek in dit proefschrift toont aan dat de voorgestelde modellen inderdaad in staat zijn om complexere onderliggende distributies vast te leggen dan wat mogelijk was met de exponentiële distributie. Bovendien worden twee methoden voorgesteld voor het schatten van parameters voor deze modellen uit volledige data en worden deze experimenteel met elkaar vergeleken.

Het experimentele onderzoek richt zich op de studie van de mogelijkheden van dynamische Bayesiaanse netwerken en continue-tijd Bayesiaanse netwerken om dynamische systemen te modelleren. In dit onderzoek worden medische praktijkgegevens gebruikt om verschillende aspecten van Bayesiaanse-netwerk-representaties te onderzoeken waar potentieel rekening mee moet worden gehouden. Deze studies werpen

licht op de eisen voor het gebruik van deze twee temporele modellen voor praktische doeleinden.

ACKNOWLEDGMENTS

This thesis is the result of the research I have been working on in the past four years. Clearly, completing this thesis would not have been possible without the help and the support from many people. Here I would like to show my gratitude to everyone who directly and indirectly contributed to the work and the final result as layed down in the thesis.

This research has been made possible by the China Scholarship Council, by a grant from the project NanoSTIMA, which was funded by the North Portugal Regional Operational Programme, under the Portugal 2020 Partnership Agreement and through the European Regional Development Fund, and by the COPD+ project supported by EFRO.

I would like to express my sincere gratitude to my promoter Peter Lucas. His invitation of starting doing a PhD in the Netherlands and coming to Radboud University, opened my eyes to a different way of combining computer science and mathematics. His immense medical knowledge and in-depth insight into medical problems helped me overcome numerous obstacles I faced in the interdisciplinary research between computer science and medicine. In particular, I am thankful for his continuous patience and support of my work in a different culture during the past four years, and in completing this thesis in time.

I also owe my gratitude to my co-promoter, Arjen Hommersom and Maarten van der Heijden, for arousing my enthusiasm and passion for academical research. Arjen, I especially enjoyed our brainstorming discussions about problems I encountered in the research, which helped me to quickly pinpoint the essential aspects of the problems. Meanwhile, you also encouraged me to pursuit my own vision and to follow my instinct through the entire PhD study. Maarten, our research collaboration was enjoyable and fruitful. After you moved away from academia, your follow-up support for the COPD project gained us a award in a Chinese competition. At the time of writing this thesis, your invaluable comments and tips also further improved it.

I spent almost half a year at the University of Milano-Bicocca, Milan, Italy, working closely with Fabio Stella, funded by a grant from the Italian AI society. I was inspired by his dedication to both theoretical and practical research in the field of continuous time Bayesian networks,

which only few researchers in the world work on. Our regular chit-chats next to the vending machine made my stay efficient as well as fun.

Working together on the My Mobile and Smart Health Care Assistant (MoSHCA) and the SensiStep projects, with a variety of people with a wide range of interests and backgrounds, was a very rewarding experience. In particular, I thank Lonneke Boer and Marco Raaben for doing an excellent job by providing me with valuable medical data for analysis. I am so grateful for the collaboration with the other members of these projects: Erik Bischoff, Leon Derks, Marleen Germs, Taco Johan Blokhuis and Albert Pla Planas.

Also, many other colleagues made my work at Radboud University enjoyable. I thank Marcos for our interesting discussions at the beginning of my PhD. Markus, I especially thank for his programming expertise, which was essential in implementing some of the experiments in the last chapter of this thesis. Aside from work, you filled my spare time with many entertaining and energy-consuming activities, in particular doing power workouts and playing chess. Paul, your help with my English when I just started my PhD proved to be invaluable, just like our friendship. Thanks Steffen, Giso and Marcos for sharing the office with me for the last years of my PhD time. Greg, Bogi, Anna, Niels, Rick, Martijn, Jurrien, David, Tim, Alexis, Joshua, Petra, Ramon, Gabriele, Daniel and everyone else I might forget, I thank you all for the great time we had. I also want to thank Ingrid for her support with organisational issues and her tips for living in a different culture.

I am also grateful for the support of my family and friends. My parents, I owe thanks to for supporting me studying abroad and to my siblings for taking care of the family. I am also lucky to have my partner and the best friend, Sven, for his listening ear, open mind, and more importantly, unconditionally support, whenever they are needed. Thanks also go to his family for making my stay in a foreign country a bit like at home.

Finally, I want to thank a group of friends I made in the Netherlands. Especially I owe my thanks to Mingjuan, for being a good friend for years. Xiaolan, Mingmei, Yanan, Lutao, Yanfen, Jiangyan, Zhangmin, Hetao, Yanzi, Liping, Xinping, Zhangyang, Wangcheng, Ruifei, Zhuoran, Zhengyu, Xuewei and many other friends I may possibly forget, your help and company makes life in the Netherlands more fun.

CURRICULUM VITAE

Manxia Liu

October 05, 1987 Born in Zhejiang, China

September 2006 - June 2010 Bachelor of Software Engineering
College of Software
Hunan University, Hunan, China

September 2010 - June 2013 Master of Software Engineering
College of Computer Science and Electronic Engineering
Hunan University, Hunan, China

October 2013 - August 2018 Doctor of Philosophy
Software Science
Institute for Computing and Information Sciences
Radboud University, Nijmegen, the Netherlands