

UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Informatica Sistemistica e Comunicazione
Dottorato di Ricerca in Informatica - XXIII Ciclo



Graphical Models for Continuous Time Inference and Decision Making

Elena Gatti
Ph.D Thesis

Supervisor: Prof. Fabio Antonio STELLA
Tutor: Prof. Carlo BATINI

Ph.D. Coordinator: Prof.ssa Stefania BANDINI

ANNO ACCADEMICO 2009/2010

*Ai miei nipoti,
ai quali auguro di fare scelte
che li rendano liberi ed orgogliosi.*

Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision of Prof. Fabio Stella during the Ph.D. curse. I thank him for the technical discussions on probabilistic graphical models, and for the valuable speeches. I wish to thank Prof. Enrico Fagiuoli and Prof. Carlo Batini for their support in the Ph.D and for feedbacks.

A particular thanks to Prof. Finn Jensen and MI group of the University of Aalborg that gave me the opportunity to work with on an interesting research topic.

At DISCo, I have been privileged to discuss ideas with a number of people. I am especially grateful for technical support and for his friendship from Davide. I am grateful for many influential conversations with Paolo, Dario, Daniela, Yuri. I would also like to acknowledge Mauro and Stefano for their amusing encouragements. I wish to thank the people working at DISCo who helped me during my PhD.

I am grateful to all my friends especially I owe a special thank to Giacomo, because he is my favorite listener! I thank Debora because she lighted my Danish period.

Finally, I am forever indebted to my parents for their understanding and endless patience. I am ever grateful to Gabriele who believed in me and encouraged when it was most required.

Contents

1	Introduction	9
1.1	Motivation	9
1.2	Overview	11
1.3	State of art	12
1.3.1	Temporal Reasoning	12
1.3.2	Probabilistic Approaches	13
1.3.3	Event history analysis	16
1.3.4	Markov models	16
1.3.5	CTBN context	17
1.3.6	Temporal information systems in Medicine	18
2	Background of CTBNs	20
2.1	Bayesian Networks	20
2.2	Dynamic Bayesian Networks	23
2.2.1	Inference in DBN	24
2.3	Continuous Time Markov Processes	26
2.3.1	Representation	27
2.3.2	Query of Markov Process	29
3	CTBN model	31
3.1	Conditional Markov Processes	32
3.2	The continuous time Bayesian network model	33
3.3	Amalgamation semantic	35
3.4	Generative semantics	36

4	Inference in Continuous Time Bayesian Networks	38
4.1	Queries and exact inference over CTBN	39
4.2	Approximate inference: Expectation Propagation	42
4.2.1	Expectation propagation for segments	43
4.3	Stochastic simulation	44
4.3.1	Forward sampling for CTBN	44
4.3.2	Gibbs Sampling for CTBN	46
5	CTBN toolbox	49
5.1	The Matlab CTBNs Environment	49
5.1.1	Model file	51
5.1.2	Evidence file	52
5.1.3	Inference engines	54
5.1.4	Output interfaces	56
5.1.5	Performance analysis	57
6	A CTBN model for Cardiogenic Heart Failure	59
6.1	Introduction	59
6.2	The Cardiogenic Heart Failure	61
6.2.1	Heart failure in the medical domain	62
6.2.2	Time granularity	63
6.2.3	The CTBN model for cardiogenic heart failure	66
6.2.4	Inference	70
6.3	Discussion	75
6.4	Conclusions	77
7	Decision Making under Uncertainty	79
7.1	Decision Theory	80
7.2	Structured Decision Problems: Decision Trees	82
7.3	Structured Decision Problems: Influence Diagrams	84
7.4	Information enhancement	87
7.4.1	Introduction	87
7.4.2	Information enhancement approach	91
7.4.3	Maximize uncertainty	93
7.4.4	Border and Frontier	95

7.4.5	Finding the border and the frontier	97
7.4.6	Cut sets between frontier and border	99
7.4.7	Cut set heuristics	100
7.4.8	Irregular networks	102
7.4.9	An iterative procedure	102
7.4.10	Conclusion and future work	102
8	Conclusion	103

List of Figures

2.1	An example of Bayesian Network	21
2.2	a) Simple DBN model. Labeled circles within a dotted oval represent our variables in one time slice. Formally, arcs connecting variables from one time slice to variables in the next have the same meaning as in a BN, but they intuitively carry a stronger implication of causality. b) Example of unrolled DBN.	25
2.3	The main types of inference. The shaded region is the interval that contains evidence. The arrow represents the time of inference. t is the current time, and T is the sequence length	26
3.1	An example of CTBN over three variables: X, Y, Z	34
4.1	Posterior of X, Y , and Z over time for the CTBN of Example 3.1	41
5.1	Drug effect network	50
5.2	CTBN model file: node's name and their cardinality for the CTBN shown in Figure 5.1	51
5.3	CTBN model file.	53
5.4	CTBN evidence file.	54
5.5	Concentration posterior probability of both states of the node.	56
5.6	Toy heart disease network	57
6.1	CTBN model for the acute myocardial infarction.	68
6.2	Posterior for AMI , UO and CardEnz under Scenario 1. . . .	72

6.3	Posterior for Pump and Shock under Scenario 1.	73
6.4	Posterior for AMI , Pump , CardEnz and Shock under Scenario 2.	74
6.5	Posterior for AMI , CardEnz and Shock under Scenario 3. . .	75
7.1	Graphical representation of a decision problem with A, B, C chance nodes. The variable D is a decision variable. The variable U is the utility node.	81
7.2	The figure depicts an ID with four chance nodes A, B, C and E, four decision nodes D_1, D_2, D_3, D_4 and three value nodes V_1, V_2 and V_3	84
7.3	An influence diagram over variables with ten states.	88
7.4	The general situation. You are in the middle of a series of decisions (D_7); you have collected much information, and now you have to anticipate a future decision (D_{10}).	90
7.5	A POMDP	92
7.6	C_6 is enhanced for D_6 . With the past of D_3 instantiated, the largest policy domain contains four variables.	92
7.7	C_5 is enhanced for D_5 . Now the largest domain contains three variables.	93
7.8	The ID in Figure 7.4 with the nodes 11 and 14 enhanced. . .	94
7.9	The ID marked after the search algorithm. "D" indicates that the node cannot be enhanced; "I" indicates nodes with information to transmit; "B" indicates border; "F" indicates frontier.	98
7.10	The free graph for the ID in Figure 7.4	99
7.11	The extended free graph for the ID in Figure 7.4	100
7.12	A junction tree for the extended free graph in Figure 7.11. . .	101

List of Tables

5.1	Table of the performance analysis	58
6.1	An example of a Conditional Probability Table in a DBN. . .	65
6.2	An example of a Conditional Intensity Matrix for the node B in a simple CTBN.	66
6.3	Nodes and states for the acute myocardial infarction CTBN.	70

Chapter 1

Introduction

1.1 Motivation

Change is something ubiquitous. We can reason about change during a series of actions and events, such as playing or investing; understanding how to express occurrence of events and how to predict their effects is a central issue in temporal reasoning. The time and nature of any particular change is influenced by many factors which are, themselves changing. Changes could be unusual and uncertain, indeed predicting the future and understating the past are challenging tasks. Particular interest can be addressed to making decision in presence of uncertainty or to planning and diagnosis dynamic domains where changes occur and uncertainty is involved. A typical situation where uncertainty comes into play is when the exact state transition dynamics are unknown and only limited or no expert knowledge is available and/or affordable. In order to better underline the importance of taking decision under uncertainty, consider real-life artificial intelligence applications that need an efficient way to reason about time. These applications include robotics, scheduling, intensive care monitoring systems, telephone networks and troubleshooting. For example, consider a particular clinical scenario in which a patient is like a complex and dynamic system, the variables that compose the system are the conditions of the patient, a clinician does not know all the states of the variables, so he acts in presence of uncertainty. In particular, suppose to know the state of few

observable variables, like the values of blood pressure or heart frequency, the clinician has to decide the best treatment or consider the probability of having a myocardial infarction. He has to answer the following questions: what kind of support does he/she can use? What kind of probabilistic models can he/she consider to solve the decision problem? All these questions involve the concept of probabilities' distributions over time. There are a lot of probabilistic methods able to model dynamic systems. This dissertation places in this research topic and in particular it concerns the inference problems and decision making mechanism in the framework of the Continuous Time Bayesian Networks, an extension of the Bayesian Networks [Pearl, 1988] proposed by Nodelman [Nodelman et al., 2002]. Modeling, learning and reasoning about these complex dynamic systems is an important task and a great challenge.

Medical tasks, such as diagnosis and therapy, are by nature complex and not easily amenable to formal approaches. Medical knowledge is inherently uncertain and incomplete. Likewise, patient data are often ridden with uncertainty and imprecision, showing serious gaps. In addition, they could be too voluminous and at a level of detail that would prevent direct reasoning by a human mind. The computer-based performance of medical tasks poses many challenges. As such it is not surprising that Artificial Intelligence researchers were intrigued with the automation of medical problem solving from the early days of Artificial Intelligence. The technology of expert systems is largely founded on attempts to automate medical expert diagnostic reasoning [Combi et al., 2010]. Physicians and other medical professionals, are required to perform various task such as diagnose the cause of a problem, predict its development, prescribe treatment, monitor the progress of a patient and overall manage a patient. The biggest challenge was the modeling of knowledge for the purpose of supporting tasks such as diagnosis, therapy and monitoring. In all cases time considerations arise. Time plays a major role in medical information systems: indeed, it is an important and pervasive concept of the real world and needs to be managed in several different ways: events occur at some time points, certain facts hold during a time period, and temporal relationships exist between facts and/or events [Ozsoyoglu and Snodgrass, 1995]. Time has

to be considered when representing information within computer-based systems, when querying information about temporal features of the represented real world and when reasoning about time-oriented data [Montanari and Pernici, 1993].

1.2 Overview

In this chapter, we review the state of art concerning the probabilistic graphical models and the methods to model, learn and reason about structured stochastic processes.

The next chapter is about the background on Bayesian networks, dynamic Bayesian networks, and continuous time Markov processes.

In Chapter 3, the Continuous Time Bayesian Network framework is presented, covering representation and semantics. It begins with the definition of conditional Markov processes, which form the building block of CTBNs. It then defines the CTBN representation and gives two alternative semantics.

Chapter 4 contains notions regarding the inference task in CTBN; both exact and approximate inference, together with stochastic simulation have been presented.

The contribution of this thesis begins with Chapter 5, covering the designed and implementation of the CTBN toolbox in Matlab. A new syntax for the specification of the model has been described. The proposed software is unique and provide a powerful tool in order to make inference with this probabilistic framework.

Chapter 6 shows how Continuous Time Bayesian Networks can be used to diagnose the cause of cardiogenic heart failure and to anticipate its likely evolution. Three case studies related to cardiogenic heart failure are presented. The model predicts the occurrence of complicating diseases and the persistence of heart failure according to variations of the evidence gathered from the patient. Predictions are shown to be consistent with current pathophysiological medical understanding of clinical pictures [Gatti et al., 2011].

Chapter 7 provides a powerful and general overview of the decision theory with particular interest on the structured decision problems. The framework of the Influence Diagrams has been used in order to propose a new approach for approximate representation of optimal strategies [Jensen and Gatti, 2010].

Chapter 8 concludes the thesis with a summary and further research directions.

1.3 State of art

1.3.1 Temporal Reasoning

The formalization of temporal reasoning proved to be a great task as the literature in logic, philosophy, linguistics, Artificial Intelligence and Computer Science testifies. There are then many perspectives and many options to consider at the stage when one is trying to define what the basic layer of a temporal system should be. We are not aiming here at covering all the aspects and will instead focus very shortly in a subset of them from the perspective of medicine. The reader is addressed to [Chittaro and Montanari, 2000] [Augusto, 2001] for lengthier descriptions of the basic concepts as well as a description of the main trends of temporal reasoning in AI [Augusto, 2005]. The explicit representation and reasoning about time is an important problem in many areas of artificial intelligence. Over the last 10-15 years, it has been attracting the attention of many researchers. Several temporal reasoning systems, differing in design issues related to ontology of time, underlying temporal logic, temporal constraints used and algorithms employed, have been developed [Pani et al., 2001]. The literature refers to the Situation Calculus [Hayes and McCarthy, 1969], a first-order predicate calculus theory for reasoning about change and some extended variants in continuous time [Pinto and Reiter, 1995]. In addition there is the McDermott's theory [McDermott, 1982] which is based on first order logic and uses discrete time. An important element of this formalism is a tree structure called chronicle tree. It represents different possible futures as branches. The tree establishes a relation between the

logic and possible world semantics. In the topic of the temporal reasoning an important contribution had been proposed by Allen [Allen, 1984]. He proposed an interval-based temporal logic theory; the formalism defines a logic for events and actions but it cannot represent uncertain events or events with probabilistic consequences. The temporal reasoning problem can also be formulated as a *constraint satisfaction problem* [Beek and Cohen, 1990]. Allen's relationships represent the possible constraints on time intervals. Temporal knowledge allows finding relationships between intervals. The problem of identifying the temporal relationships between intervals is called the minimal labels problem. The problem is NP-complete [Vilain et al., 1989] and heuristic algorithms can solve the constraint satisfaction problem. This representation has been implemented in systems such as *Tachyon* [Stillman et al., 1993] and is particularly popular for scheduling applications [Henke, 1994].

1.3.2 Probabilistic Approaches

The next paragraphs deal with probabilistic representation of uncertainty, emphasizing both its strong theoretical basis and its possibility to be applied to the study of phenomena in fields such as engineering, management science, the physical and social sciences, and operational research. Any realistic model of a real-world phenomenon must take into account the possibility of randomness. That is, more often than not, the quantities we are interested in will not be predictable in advance but, rather, will exhibit an inherent variation that should be taken into account by the model. This is usually accomplished by allowing the model to be probabilistic in nature. Such a model is, naturally enough, referred to as a probability model [Ross, 2007]. Probabilistic reasoning systems reason with uncertainty according to the laws of probability theory. The main advantage of such systems over systems that use qualitative logics for inference is that they are able to reach conclusions even if the available information is not enough to prove any conclusion. The inability of qualitative logic to deal with real world imprecision and uncertainty is sometimes called brittleness. The underpinning of the probabilistic approaches is the Bayes' theorem. It forms the fun-

damental tool for belief revision, and ‘Bayesian networks’ can be formed by superimposing a probability model on a graph representing qualitative conditional independence assumptions. The resulting structure is capable of representing a wide range of complex domains [Cowell, 1999]. There are many good reasons to choose probabilistic networks as the framework for solving inference and decision problems under uncertainty. As indicated above, these include:

1. Coherent and mathematically sound handling of uncertainty,
2. Normative decision making,
3. Automated construction and adaptation of models based on data,
4. Intuitive and compact representation of cause-effect relations and (conditional) dependence and independence relations, and
5. Efficient solution of queries given evidence [Kjaerulff and Madsen, 2007].

In the probabilistic approaches, uncertainties are characterized by the *probabilities* associated with *events*. An event corresponds to any of the possible states a physical system can assume, or any of the possible predictions of a model describing the system. For example in the study of environmental pollution, the situation where the contaminant concentration exceeds a regulatory level can be an event. Similarly, in case of mathematical modeling, the situation where the model outputs fall in a certain range, is also an event. In short, any possible outcome of a given problem, such as experimental measurement of contaminant concentration or mathematical modeling of a physical system, is an event. The *probability* of an event can be interpreted in terms of the frequency of occurrence of that event. When a large number of *samples* or *experiments* are considered, the probability of an event is defined as the ratio of the number of times the event occurs to the total number of samples or experiments. Examples of experiments or samples include a large number of independent measurements of contaminant concentration, or a large number of simulations using a mathematical

model with randomly varying parameters. All these aspects touch on a wide variety of issues; they are all questions about distributions of events over time. Standard ways of approaching such questions like event history analysis [Blossfeld and Hamerle, 1992, Blossfeld and Rohwer, 2002] and Markov process models [Aalen, 1975, Andersen and Borgan, 1985, Duffie et al., 1996, Lando, 1998] work well, but there is a limit regarding the dimension of the state space. A number of methodologies have been developed to handle temporal reasoning using the Bayesian Networks, Tawfik and Neufeld [Tawfik and Neufeld, 1994] proposed Temporal Bayesian Networks which allow time to be modelled both discrete or continuous. Probabilities are represented as functions of time. With this type of probabilistic framework is possible only to calculate the distributions over the state at a given time and not distributions over time. Arroyo-Figueroa and Sucar [Arroyo-Figueroa and Sucar, 1999] present Temporal Nodes Bayesian Networks (TNBNs) this approach is based on time intervals; the events can only happen in one of the intervals. Tawfik and Neufeld [Tawfik and Neufeld, 2000] give an overview of techniques used for modelling time with Bayesian networks.

Unfortunately, including more factors in such models comes at the price of an exponential increment in the size of the state space. The explosion of the state space makes the reasoning task more difficult. New approaches based on continuous time models over structured state space need to be introduced. A section of the state of art will include also the overview of the CTBN framework. The approach is based on the framework of homogeneous Markov processes, but utilizes ideas from Bayesian networks to provide a graphical representation language for these systems. In particular, the graphical structure encodes conditional independences in the distribution over the evolution of the system. Moreover, this framework explicitly represents temporal dynamics and allows us to query the network for the distribution over the time when particular events of interest occur [Nodelman et al., 2002].

1.3.3 Event history analysis

Event history analysis is a term commonly used to describe a variety of statistical methods that are designed to describe, explain or predict the occurrence of events. Outside the social science, these methods are also called survival analysis, owing to the fact that they were originally developed by biostatisticians to analyze the occurrence of deaths. But despite their biomedical origin, these same methods are perfectly suitable for studying a vast array of social phenomenas such births, marriages, divorces, job terminations, promotions, arrests, migrations and revolutions. There is a broad class of related methods we refer to as event history analysis though other terms - e.g., survival analysis, hazard models, failure analysis, duration analysis, etc. [Blossfeld et al., 1989] [Andersen et al., 1993]. In general an event can be defined as qualitative change that occurs in some particular point in time. To apply event history methods you need event history data - a longitudinal record of when events occur to some individuals or sample of individuals. For example if you ask a human resource office to report the job terminations dates of all the employers, you will get a set of event history data that will allow you to analyze the occurrence of job terminations. Of course if you want to do causal or predictive analysis, you would also want to measure possible explanatory variables. Some of these may be constant over time while others may vary with time [Allison, 2004] [Yamaguchi, 1991].

The purpose of event history analysis is to explain why certain individuals are at a higher risk of experiencing the event(s) of interest than others.

1.3.4 Markov models

The Markov process models are a class of methods used for continuous time modeling. In [Aalen, 1975, Andersen et al., 1993] the counting process model has been presented; specifically, the event can be repeated and the focus is to calculate the distribution over how many times the event has occurred. In the well known queueing theory [Lipsky, 1992] the events are two, denoted with 'arrival to' and 'departure from' the queue. The amount of time between the arrival and the departure events can be computed.

1.3.5 CTBN context

This paragraph concerns the overview of the CTBN framework, the contributions of researchers in the last few years will proposed in order to better underline the segment of research area in which this dissertation is introduced.

Recently, Nodelman et al. [Nodelman et al., 2002] presented continuous time Bayesian networks (CTBNs), which provide a structural representation of continuous-time, finite-state, Markov processes. A CTBN factorizes a multi-variable Markov process into local variables using a graphical representation. Each variable is modeled as an inhomogeneous Markov process whose dynamics depend on the current value of a set of variables in the system. Since we use a continuous-time model, each variable is an inhomogeneous Markov process. The dynamics of each variable depend on the current instantiation of a set of other variables in the model. These dependencies are represented using the arcs in the graph. By doing this, CTBNs explicitly represent the temporal dynamics in continuous time and exploit the dependencies among stochastic variables using a structured representation. Thus, exponential explosion in the representation is avoided, and queries can be answered using distributions over continuous time. Because of these advantages, CTBNs have been applied to various real world systems, including human-computer interactions [Nodelman and Horvitz, 2003], server farm failures [Herbrich et al., 2007], robot monitoring [Ng et al., 2005], network intrusion detection [Xu and Shelton, 2008], and social network analysis [Fan and Shelton, 2009]. Kan and Shelton [Kan and Shelton, 2008] exploited the CTBN representation and presented an approach to solve structured continuous-time Markov decision processes.

The key idea is that a central task in many applications is reasoning about processes that change over continuous time, in this direction *Continuous Time Bayesian Networks* are a general compact representation language for multi-component continuous-time processes. However, exact inference in such processes is exponential in the number of components, and thus infeasible for most models of interest. Inference for CTBNs is the task of estimating distributions given some observations and Nodelman [Nodelman

et al., 2005] proposed an algorithm that uses message-passing in a cluster graph, a variant of the Expectation Propagation algorithm. It takes computational advantage of the conditional independences and the explicit temporal model. Nodelman provided algorithms for learning CTBN [Nodelman et al., 2003] parameters from complete data, including maximum likelihood and Bayesian approaches. Gopalratnam showed how to model arbitrary transition time distributions in CTBNs using Erlang-Coxian approximations, while maintaining tractable learning [Gopalratnam et al., 2005]. He suggested a general method for reasoning about negative evidence, by introducing updates that assert no observable events occur over an interval of time. He demonstrated that interval evidence updates can significantly improve the accuracy of filtering and prediction.

The CTBN framework and its algorithms for approximate inference have been deeply used by El-hay et al. [El-Hay et al., 2006]. They proposed an alternative representation language Continuous Time Markov Networks that represents a different type of continuous-time dynamics, particularly appropriate for modeling biological and chemical systems. Sampling procedures have been implemented by Fan and Shelton [Fan and Shelton, 2008, Fan et al., 2010]. They presented a sampling algorithm for continuous time Bayesian networks based on importance sampling. An extension to continuous time particle filtering and smoothing algorithms has been proposed. The three algorithms can estimate the expectation of any function of a trajectory, conditioned on any evidence set constraining the values of subsets of the variables over subsets of the time line. Gibbs sampling procedure has been applied for the inference of CTBN by El-Hay et al. [El-Hay et al., 2008]. El-Hay developed a novel Gibbs sampling procedure for multi-component processes. This procedure iteratively samples a trajectory for one of the components given the remaining ones.

1.3.6 Temporal information systems in Medicine

In the medical-informatics field, temporal data modeling, temporal maintenance, and temporal reasoning have been investigated, to support both electronic medical records and medical decision support systems [Kohane,

1987] [Kahn et al., 1991] [Combi et al., 1995]. It is important to provide a kind of definition for the meaning we assign to the expression 'temporal information system': *temporal information systems in medicine are information systems able to store, manage, query, and support different inference tasks on time-oriented clinical data*. Temporal information systems in medicine can be distinguished from the technical tasks, they concern with. For example, there are applications in:

- managing of time-oriented data stored in medical records of ambulatory or hospitalized patients;
- predicting of future values of clinical data, given past trends;
- abstracting of time-oriented clinical data;
- time-oriented knowledge-based decision support, such as for systems supporting diagnosis, monitoring , or therapy planning [Chittaro and Montanari, 2000] ;

There is a second important perspective concerning the *medical tasks* that need to deal with temporal aspects of clinical data. The literature proposed the following systems:

- diagnosis [Bichindaritz and Conlon, 1996, Keravnou and Washbrook, 1990];
- therapy administration and monitoring [Kahn and Marrs, 1995];
- protocol and guideline-based therapy [Das and Musen, 1994];
- patient management [Combi et al., 1995].

Chapter 2

Background of CTBNs

2.1 Bayesian Networks

A Bayesian network (BN) [Pearl, 1988], also known as a belief network or a Bayesian belief network, is a graphical model for probabilistic relationships among a set of variables. For over a decade, expert systems use BNs in domains where uncertainty plays an important role. Nowadays, BNs appear in a wide range of diagnostic medical systems, fraud detection systems, missile detection systems, etc. BNs can handle incomplete data sets and allow one to learn about relationships between variables; it is possible to combine expert knowledge and data into a BN. Using Bayesian methods, the overfitting of data during learning can be avoided relatively easily. Furthermore these graphical models are able to model causal relationships between variables. BNs have a qualitative and a quantitative component. The qualitative component is represented by the network structure and the quantitative component is expressed by the assignment of the conditional probability distributions to the nodes. A Bayesian network is a graphical model for a joint distribution of random variables. The qualitative part is defined as an acyclic directed graph (DAG) whose nodes represent random variables and edges represent dependencies among variables. The nodes in the BN represent discrete or continuous random variables. If an arc points from node A to C then A is a parent of C and C is a child of A . A parent directly influences its children. The parents of a generic node X are

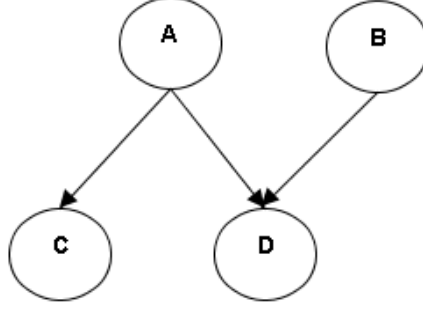


Figure 2.1: An example of Bayesian Network

denoted by $Pa(X)$. Furthermore, every node has its own local probability distribution. All these components together form the joint probability distribution (JPD) of the BN, the so called quantitative component of the BN. The joint probability distribution of a generic node X follows from applying the chain rule on the network structure:

$$p(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | Pa(X_i)) \quad (2.1)$$

Indeed, the joint distribution of the variables is specified by describing prior probabilities for the root nodes and conditional probabilities for each variable that is not a root of the graph. For example, in the network of Figure 2.1 one would need the conditional probabilities $P(C|A)$ and $P(D|A, B)$ and the prior probabilities $P(A)$ and $P(B)$. The chain rule of Bayesian networks states that the joint distribution of the random variables is given by the product of the conditional probabilities, i.e., $P(A, B, C, D) = P(A)P(B)P(C|A)P(D|A, B)$ [Jensen and Nielsen, 2007].

Bayesian networks are used for calculating new probabilities when you get new information, called *evidence*. The evidence $\mathbf{e}$ is expressed as an instantiation of some variables in the BN. Given a generic node X and the evidence $\mathbf{e}$ $p(X|\mathbf{e})$ needs to be calculated. The two rules that are needed for inference are Bayes' theorem and the expansion rule (2.1). Inference can be best explained by an example. In the network of Figure 2.1 variables A and B are independent if the state of D is not observed. Thus, without any

additional evidence, observing the state of variable A does not affect the distribution of variable B , and vice versa. Similarly, the variable C and the rest of the network are independent if the state of variable A is observed, and thus the distribution of variable C does not need to be updated after observing additional evidence. Suppose A is observed ($A = \text{true}$) and the observer wants to know what is the probability of $C = \text{true}$, given $A = \text{true}$. Thus, $P(C = \text{true})$ needs to be calculated. C is called a *query variable* and A is called an *evidence variable*. The calculation comes down to:

$$p(X|e) = p(C|A) = \frac{p(A|C) \cdot P(C)}{p(A)}. \quad (2.2)$$

Exact inference of large, multiply connected BNs becomes very complex. In fact, it is NP-hard [Cooper, 1990]. Because of this intractability, approximate inference methods are essential. In general, randomized sampling algorithms, also called Monte Carlo algorithms, are used.

For exact inference, variable elimination algorithms or junction tree algorithms can be used. Variable elimination uses a smart way to rewrite the inference calculations by choosing a specific elimination order of the variables, making computation more efficient. Junction tree algorithms convert the multiply connected BNs to junction trees, after which inference can be performed using variable elimination (Shafer-Shenoy algorithm) [Shafer and Shenoy, 1988] or belief propagation [Lauritzen and Spiegelhalter, 1988]. Of course, this conversion can be very complex as well, so it is not suitable for every BN. A third technique that can be used to reduce complexity of the inference computations is called relevance reasoning [Druzdzel and Suermondt, 1994]. This is a preprocessing step that explores structural and numerical properties of the model to determine what parts of the BN are needed to perform the computation. Relevance reasoning reduces the size and the connectivity of a BN by pruning nodes that are computationally irrelevant to the nodes of interest in the BN.

Special algorithms that take advantage of conditional independences have been developed for probability updating. There exists numerous computer software using these algorithms for efficient creation and analysis of Bayesian network models, e.g., Hugin and GeNIe [Hug, 2010] [Gen, 2010].

2.2 Dynamic Bayesian Networks

The Bayesian Network framework is a very efficient language for modeling static problem domains. In such a situation, every variable has a single and fixed value. Unfortunately, this static assumption is not always sufficient, as many examples of the real life show. Dynamic Bayesian Networks (DBNs) are directed graphical models of stochastic processes. They generalize hidden Markov models (HMMs) and linear dynamical systems (LDSs) by representing the hidden (and observed) state in terms of state variables, which can have complex interdependencies. The graphical structure provides an easy way to specify these conditional independences, and hence to provide a compact parameterization of the model. It is assumed that the model structure in a DBN does not change, but the term DBN has become entrenched. We also normally assume that the parameters do not change, i.e., the model is time-invariant. A *Dynamic Bayesian Network* (DBN), which is the natural extension of the BN framework, used to model time-dependent phenomena [Dean and Kanazawa, 1989]. A DBN is a BN composed of time slices, each of which describes the variables and conditional probabilities at a given time instant, and temporal links between the time slices. Obviously the direction of the temporal links is set to be consistent with the direction of time. A dynamic Bayesian network (DBN) is a way to extend Bayes nets to model probability distributions over semi-infinite collections of random variables, $X_1, X_2, \dots, X_n$. We only consider discrete-time stochastic processes, so we increase the index t by one every time a new observation arrives. [Murphy, 2002] (The observation could represent that something has changed making this a model of a discrete-event system.) Note that the term ‘dynamic’ means we are modeling a dynamic system, not that the network changes over time. A DBN is defined to be a pair, $(B_1; B \rightarrow)$, where B_1 is a BN which defines the prior $P(X_1)$, and $B \rightarrow$ is a two-slice temporal Bayes net (2TBN) which defines $P(X_t|X_{t-1})$ by means of a DAG (directed acyclic graph) as follows:

$$P(X_t|X_{t-1}) = \prod_{i=1}^N P(X_t^i | Pa(X_t^i)) \quad (2.3)$$

where X_t^i is the ith node at time t , and $Pa(X_t^i)$ are the parents of X_t^i in the graph. The nodes in the first slice of a 2TBN do not have any parameters associated with them, but each node in the second slice of the 2TBN has an associated conditional probability distribution (CPD), which defines $P(X_t^i | Pa(X_t^i))$ for all $t > 1$. The parents of a node, $Pa(X_t^i)$, can either be in the same time slice or in the previous time slice, i.e., we assume the model is first-order Markov. However, this is mostly for notational simplicity: there is no fundamental reason why we cannot allow arcs to skip across slices. The arcs between slices are from left to right, reflecting the causal flow of time i.e. see Figure 2.2. The arcs within a slice are arbitrary. Intuitively, directed arcs within a slice represent ‘instantaneous’ causation. It is also useful to allow undirected arcs within a slice, which model correlation or constraints rather than causation; the resulting model is then called a (dynamic) chain graph.

- Intra-time-slice arcs are allowed between the variables within a time slice.
- Inter-time-slice arcs are allowed from a variable in one time-slice to any variable. in the next time slice

The semantics of a DBN can be defined by ‘unrolling’ the 2TBN until we have T time-slices. The resulting joint distribution is then given by

$$P(X_{t:T}) = \prod_{t=1}^T \prod_{i=1}^N P(X_t^i | Pa(X_t^i)). \quad (2.4)$$

2.2.1 Inference in DBN

A graphical model specifies a complete joint probability distribution (JPD) over all the variables. Given the JPD, we can answer all possible inference queries by marginalization (summing out over irrelevant variables). Assuming to have n nodes, with cardinality 2, the JPD has size $O(2^n)$ and hence summing over the JPD takes an exponential time. Some efficient methods have been proposed in literature for this kind of task, for example variable elimination or dynamic programming.

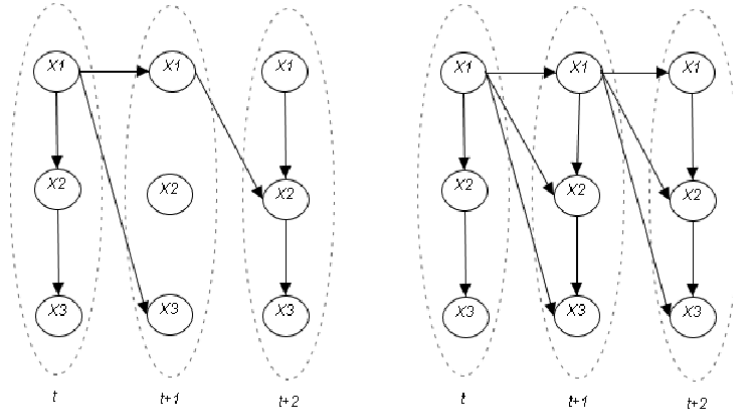


Figure 2.2: a) Simple DBN model. Labeled circles within a dotted oval represent our variables in one time slice. Formally, arcs connecting variables from one time slice to variables in the next have the same meaning as in a BN, but they intuitively carry a stronger implication of causality. b) Example of unrolled DBN.

The most common types of inference are given in Figure 2.3 and discussed next.

- With filtering (or monitoring), the current belief state is computed given all evidence from the past. To achieve this, $p(X_t|y_{1:t})$ needs to be calculated. Filtering is used to keep track of the current state for making rational decisions. It is called filtering, because noise is filtered out from the observations.
- With prediction, a future belief state is computed given all evidence from the past. This means $p(X_{t+h}|y_{1:t})$ needs to be calculated for some $h > 0$. Prediction can be used to evaluate the effect of possible actions on the future state.
- With smoothing (or hindsight), a belief state in the past is calculated given all evidence up to the present. So, $p(X_{t-l}|y_{1:t})$ is calculated for some fixed time-lag $\ell > 0$. Smoothing is useful to get a better estimate of the past state, because more evidence is available at time t than at time $t - \ell$. In figure 2.3, two types of smoothing are given: fixed-lag

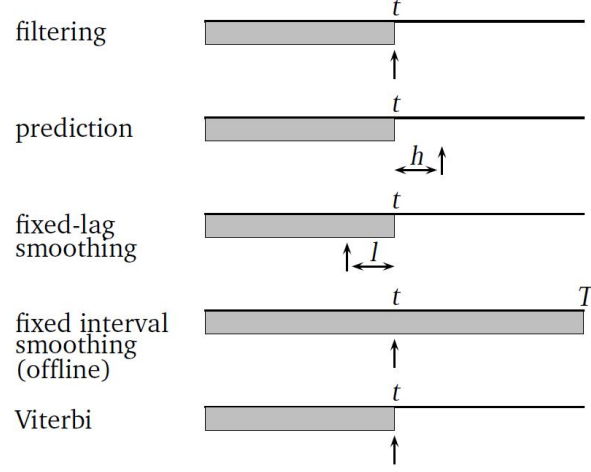


Figure 2.3: The main types of inference. The shaded region is the interval that contains evidence. The arrow represents the time of inference. t is the current time, and T is the sequence length

smoothing, which is the type of smoothing given above, and fixed-interval smoothing, which is the off-line case in which $p(X_t|y_{1:T})$ is calculated for all $1 < t < T$.

- The final inference method to be described here is Viterbi decoding (or most likely explanation). This is a different kind of inference, but it is used very often nonetheless. With Viterbi decoding, one wants to compute the most likely sequence of hidden states given a sequence of observations. That is: $\text{argmax}_{x_{1:t}} p(x_{1:t}|y_{1:t})$ needs to be calculated.

2.3 Continuous Time Markov Processes

Continuous-time Markov process allows us to realistically model and reason about these dynamic systems using continuous time. However, the size of the state space of a continuous-time Markov process grows exponentially with the number of variables in the system, which makes this method infeasible for large systems. One solution for such state space explosion is to use models with structured state spaces in which the dynamics of each local variable can be represented more efficiently. In this section,

we provide the background material about continuous-time Markov processes [Norris, 1997]. We first give the definition for a continuous-time, finite-state, homogeneous Markov process, and then discuss how to reason about continuous-time Markov processes.

2.3.1 Representation

Let X be a continuous-time, finite-state, homogeneous Markov process. X has n states $\{x_1, x_2, \dots, x_n\}$. $X(t)$ is the finite state of the system at time t . The collection of random variables $\{X(t) | t \in R_+\}$ composes the process. X satisfies the *Markov assumption*. That is,

$$P(X(t+s) = x_j | X(s) = x_i, X(r) = x_r) = P(X(t+s) = x_j | X(s) = x_i) \quad (2.5)$$

for all s, t such that $s < t < \infty$. In other words the future of a process is independent of its past given its present.

The initial distribution $P_X^0 = P(x(0))$ is a multinomial distribution over N states of X . The transient behavior of X is described by the initial distribution $P_X^0 = P(x(0))$ and the transition model which is often represented by the intensity matrix:

$$Q_X = \begin{bmatrix} -q_1^x & q_{12}^x & \cdots & q_{1N}^x \\ q_{21}^x & -q_2^x & \cdots & q_{2N}^x \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1}^x & q_{N2}^x & \cdots & -q_N^x \end{bmatrix}$$

where $q_i^x = \sum_{j \neq i}^N q_{ij}^x$, can be interpreted as the *instantaneous probability* to leave the state x_i while q_{ij}^x gives the instantaneous probability to transition from state x_i to state x_j . The diagonal elements q_i^x and the off-diagonal elements q_{ij}^x define the instantaneous transition probabilities of X . More precisely,

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} P(X(t + \Delta t) = x_j | X(t) = x_i) &= \lim_{\Delta t \rightarrow 0} q_{ij} \Delta t + O(\Delta t^2), i \neq j \\ \lim_{\Delta t \rightarrow 0} P(X(t + \Delta t) = x_i | X(t) = x_i) &= \lim_{\Delta t \rightarrow 0} 1 - q_i \Delta t + O(\Delta t^2). \end{aligned} \quad (2.6)$$

The intensity matrix Q_X is time invariant. Given Q_X , the transient behavior of X can be described as follows: X stays in state x_i for an amount of time t and transitions to state x_j . t is exponentially distributed with parameter q_i^x . That is, the probability density function f and the corresponding distribution function F for $X(t)$ remaining equal to q_i^x are given by:

$$f(t) = q_i^x \exp(-q_i^x t), t \geq 0. \quad (2.7)$$

$$F(t) = 1 - \exp(-q_i^x t), t \geq 0. \quad (2.8)$$

The expected time of transitioning is $1/q_i^x$. Upon transitioning, the probability that X transitions from state x_i to x_j is

$$q_{ij}^x / q_i^x. \quad (2.9)$$

Example 1. Assume we want to model the behavior of a computer activity $A(t)$ which has three values ($Val(A(t)) = a_o = off, a_1 = stand-by, a_2 = on$). We could represent the dynamics of $A(t)$ using the above intensity matrix:

$$Q_A = \begin{bmatrix} -1.5 & 1.2 & 0.3 \\ 0.5 & -1 & 0.5 \\ 0.8 & 2.2 & -3 \end{bmatrix}$$

If we set the time unit to one minute, this means that we expect the computer changes his status in $1/3 = 0.334$ minutes if its current status is on. When the activity is changing, with probability $0.8/3 = 0.267$ the new status will be off and with probability $2.2/3 = 0.734$ the new status will be stand-by.

Often, the system dynamics we are trying to model contain more than one variable. To model a multi-variable system, we first combine all variables into a single joint variable by ordering and enumerating all possible states of the variables. If the system has N variables X_i ($i = 1, \dots, N$), and each variable contains s_i states, the total number of states of the joint process is $n = \prod_{i=1}^N s_i$ and the size of the intensity matrix for the joint process is $n \times n$. As the number of variable increases, the size of the intensity matrix

grows exponentially.

Example 2. Assume that in the previous example, we also need to consider the effect of the electricity $E(t)$ which has two values ($Val(E(t)) = \{e_0 = stop, e_1 = running\}$). To model the dynamics of this system, we first list all the possible values of the joint variable:

$$(e_0, a_0), (e_0, a_1), (e_0, a_2), (e_1, a_0), (e_1, a_1), (e_1, a_2)$$

We then write the transition intensity of each pair of values into the joint intensity matrix:

$$\begin{array}{c} \begin{matrix} & e_0a_0 & e_0a_1 & e_0a_2 & e_1a_0 & e_1a_1 & e_1a_2 \end{matrix} \\ \begin{matrix} e_0a_0 \\ e_0a_1 \\ e_0a_2 \\ e_1a_0 \\ e_1a_1 \\ e_1a_2 \end{matrix} \begin{pmatrix} -5.5 & 1.7 & 1.3 & 2.5 & 0 & 0 \\ 1.3 & -4.3 & 1.5 & 0 & 1.5 & 0 \\ 3.9 & 1.7 & -6.5 & 0 & 0 & 0.9 \\ 2.6 & 0 & 0 & -4.9 & 2.2 & 0.1 \\ 0 & 0.25 & 0 & 0.35 & -0.75 & 0.15 \\ 0 & 0 & 2 & 1.5 & 1.5 & -5 \end{pmatrix} \end{array}$$

Notice that elements representing $E(t)$ and $A(t)$ changing simultaneously are all zeros. This means that we assume any two variables cannot change their state at exactly the same time. Since we are modeling a continuous-time dynamics, this is a natural assumption.

2.3.2 Query of Markov Process

Given the intensity matrix Q_X and the initial distribution P_0^X of a continuous time homogeneous Markov process X , there are a number of questions we can answer. Commonly we wish to calculate the conditional probability $P(X(t) = x_t | X(s) = x_s)$, where x_t is the value of X at time t and x_s is the value of X at an earlier time s . If X has n states $x_1, \dots, x_n$, all the conditional probabilities can be represented using an $n \times n$ matrix $P(X(t) | X(s))$, where the ith row vector is the conditional distribution over the value of X at time t given $X(s) = x_i$. $P(X(t) | X(s))$ can be calculated as:

$$P(X(t) | X(s)) = \exp(Q_X(t - s)). \quad (2.10)$$

Let $P_X(t)$ be a row vector representing the distribution over the values of X at time t . This distribution can be calculated as:

$$PX(t) = P_X^0 \exp(Q_X t). \quad (2.11)$$

where the initial distribution P_X^0 is represented as a row vector. The function $\exp()$ denotes the matrix exponential, which is defined as follows for a matrix Q :

$$\exp(Q) = \sum_{k=0}^{\infty} \frac{Q^k}{k!}. \quad (2.12)$$

though the matrix exponential is defined with this equation, it is not generally a good way to compute it - Moler and Loan [Moler and Van Loan, 2003]. For general real matrices, the matrix exponential can only be calculated numerically.

Chapter 3

CTBN model

Dynamic Bayesian Networks describe the state of the dynamical system at discrete time points, but do not model time explicitly. Therefore, "*since DBNs slice time into fixed increments, one must always propagate the joint distribution over the variables at the same rate*" [Nodelman et al., 2002]. Thus, if the system consists of processes which evolve at different time granularities and/or the obtained observations are irregularly spaced in time, the inference process may become computationally intractable. Continuous Time Bayesian Networks (CTBNs) [Nodelman et al., 2002] have been proposed to overcome the main limitations of DBNs. CTBNs explicitly represent temporal dynamics and allow us to recover the probability distribution over time when specific events occur. CTBNs are based on homogeneous Markov processes, while they exploit BNs to provide an intuitive language to describe complex dynamical systems. CTBNs have been used to model the presence of people at their computers together with the specific application they are using (e.g., email, word processing, web browsing, etc...) [Nodelman and Horvitz, 2003]. They have been used for dynamical systems reliability modeling and analysis [Boudali and Dugan, 2006], network intrusion detection [Xu and Shelton, 2008] and for modeling social networks [Fan and Shelton, 2009]. The semantics of CTBNs can be understood in two terms. The first is based on viewing the entire system as a composite intensity matrix describing a homogeneous Markov process over the joint entire state space via an amalgamation operation which combines all the local CIM

to produce the Joint Intensity Matrix. The evolution of the CTBN is then completely specified by this Joint Intensity Matrix. The other is based on a generative perspective where the CTBN is viewed as defining a generative model over a set of events which correspond to variables in the system taking on certain values at specific times [Nodelman, 2007].

3.1 Conditional Markov Processes

A Continuous Time Bayesian Network (CTBN) is a graphical model whose nodes are associated with random variables whose state evolves continuously over time. As a consequence the evolution of each variable depends on the state of its parents in the graph. To overcome the main limitation of Markov processes the notion of *conditional Markov process* has to be introduced. This is a particular kind of inhomogeneous Markov process, in the sense that, for any given random variable, the intensities are a function of the current values of a particular set of other variables, which also evolve as Markov processes. Therefore, intensities vary over time but not as a direct function of it. To clarify how the conditional Markov process is described, let X be a random variable whose domain is $Val(X) = \{x_1, \dots, x_k\}$ and assume that it evolves as a Markov process $X(t)$. Furthermore, assume that the dynamics of $X(t)$ are conditionally dependent from a set V of random variables evolving over time. Then the dynamics of $X(t)$ can be fully described by means of a Conditional Intensity Matrix (CIM), which can be written as follows:

$$Q_{X|V} = \begin{bmatrix} -q_1^x(V) & q_{12}^x(V) & \dots & q_{1k}^x(V) \\ q_{21}^x(V) & -q_2^x(V) & \dots & q_{2k}^x(V) \\ \dots & \dots & \dots & \dots \\ q_{k1}^x(V) & q_{k2}^x(V) & \dots & -q_k^x(V) \end{bmatrix}$$

A CIM is a set of intensity matrices, one intensity matrix for each instantiation of values v to the set of variables V . Using the BNs terminology, the variables belonging to the set V are called the parents of the random variable X . This set is usually denoted $pa(X)$, while in the case where the

parent set $pa(X)$ is empty, the CIM is simply a standard intensity matrix.

3.2 The continuous time Bayesian network model

Conditional Intensity Matrices (CIMs) allow us to model local dependencies between random variables, which is a fundamental aspect of both BNs and CTBNs. Given a set of CIMs they can be put together to obtain a single structured model which fully describes the aspects of the evolution of a multivariate probability distribution. A CTBN consists of two main components: i) an initial probability distribution and ii) the dynamics which rule the evolution over time of the joint probability distribution associated with the CTBN.

Definition 1. [Nodelman et al., 2002] Let $\mathbf{X}$ be a set of local variables $X_1, \dots, X_n$. Each X_i has a finite domain of values $Val(X_i)$. A continuous time Bayesian network $\aleph$ over $\mathbf{X}$ consists of two components: The first is an initial distribution $P_{\mathbf{X}}^0$, specified as a Bayesian network $\mathcal{B}$ over $\mathbf{X}$. The second is a continuous transition model, specified as

- A directed (possibly cyclic) graph G whose nodes are $X_1, \dots, X_n$; $pa(X_i)$ denotes the parents of X_i in G .
- A conditional intensity matrix, $Q_{X_i|pa(X_i)}$, for each variable $X_i \in \mathbf{X}$.

CTBNs allow, differently from BNs, cycles in the graph G . Indeed, an arc directed from node X to node Y in the graph implies that the dynamics of the random variable Y 's evolution in time depends on the state of the random variable X . Thus, there is no motivation why the dynamics of the random variable X 's evolution cannot simultaneously depend on the state of the random variable Y . This dependency is analogous to a DBN model where we have an arc directed from $X(t)$ to $Y(t+1)$ and an arc directed from $Y(t)$ to $X(t+1)$.

Example 3. Figure 3.1 shows the graph structure, i.e. the qualitative component, for a simple CTBN model consisting of three nodes, X , Y and Z , associated with three binary random variables. In this model we have $pa(X) = Z$, $pa(Y) = X$

and $pa(Z) = Y$ and thus a cycle such that X influences Y , Y influences Z which in turns influences X . The quantitative component of the CTBN model consists of

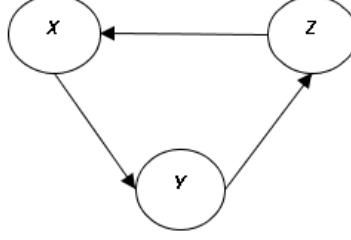


Figure 3.1: An example of CTBN over three variables: X, Y, Z

the initial distribution $P^0_{(X,Y,Z)}$ which we assume to be uniform and the following CIMs:

$$\mathbf{Q}_{X|Z=z_1} = \begin{bmatrix} -2 & 2 \\ 4 & -4 \end{bmatrix} \quad \mathbf{Q}_{X|Z=z_2} = \begin{bmatrix} -1 & 1 \\ 9 & -9 \end{bmatrix}$$

$$\mathbf{Q}_{Y|X=x_1} = \begin{bmatrix} -1 & 1 \\ 3 & -3 \end{bmatrix} \quad \mathbf{Q}_{Y|X=x_2} = \begin{bmatrix} -5 & 5 \\ 4 & -4 \end{bmatrix}$$

$$\mathbf{Q}_{Z|Y=y_1} = \begin{bmatrix} -6 & 6 \\ 8 & -8 \end{bmatrix} \quad \mathbf{Q}_{Z|Y=y_2} = \begin{bmatrix} -9 & 9 \\ 2 & -2 \end{bmatrix}$$

In the case where we view units for time as hours, the CIM $\mathbf{Q}_{X|Z=z_1}$, associated with the random variable X in the case where its parent Z is in state z_1 , means that if $X = x_1$, then we expect that X will transition to state x_2 on average in $\frac{1}{2}$ hours. While, in the case where its parent is in state z_2 , the CIM $\mathbf{Q}_{X|Z=z_2}$ means that if $X = x_2$, then we expect that X will transition to state x_1 on average in $\frac{1}{9}$ hours.

By using the graphical model, CTBNs not only provide a structured representation language for multi-variable Markov processes, but also describe the independences (and dependencies) among variables. Since a CTBN represents continuous-time dynamic systems, the independences specified by a CTBN are between distributions over entire trajectories of the variables. Similar to a Bayesian Network, each variable X_i in a CTBN is separated

from all the other variables by its Markov blanket. The Markov blanket of variable X_i is defined as the parents of X_i , the children of X_i , and the children's parents. In a CTBN, X_i is independent of all the other variables, given the entire trajectory of the Markov blanket of a variable X_i .

3.3 Amalgamation semantic

In order to describe the behavior of the CTBN we need to construct a single homogeneous Markov process. At the aim of combining the quantitative components of multiple variables, we need an operation on the CIMs. This operation is called amalgamation [Nodelman et al., 2002]. This calculation combines two CIMs to produce a single, larger CIM. The new CIM contains the intensity's taxes for the variables in S conditioned on those of C . As mentioned before, a basic assumption is that, as time is continuous, variables cannot transition at the same instant. Thus, all intensities corresponding to two simultaneous changes are zero. As we will be discussing the structure of single matrix over multiple variables, we need a function to map rows of the larger matrix to instantiations of the variables. To do this we use an ordering of the variables which, in turn, defines a unique sequence of instantiations to the set. With that, we will be able to define the expansion of a CIM (defined as a set of matrices) into a single matrix over a larger set of variables. Then we will describe the amalgamation operation which leads to the joint intensity matrix, that expresses the behavior of a CTBN as a homogeneous Markov process. If each variable X_i in the CTBN $\aleph$ has n_i states, the number of states of the joint Markov process is $n = \prod n_i$ and Q is an $n \times n$ matrix. We generate the joint intensity matrix Q by three steps:

- Let i and j be any pair of states in the joint Markov process where there is only one variable whose value is different between them. Let X be that different variable and V be the set of parents of X . Let v be the instantiation of V in state i and j . The value of the off-diagonal element q_{ij} in Q is set to be the corresponding intensity in the CIM $Q_{X|v}$.

- All the other off-diagonal elements are zero since two variables cannot transition at exactly the same time in a CTBN.
- The diagonal elements are computed to make each row sum to zero.

The mapping between row or column numbers and instantiations of the variables is provided by a variable ordering [Nodelman, 2007]. Before the amalgamation we need an expansion of the considered CIM into an expanded state space which is a joint state space. We can now define the amalgamation operation in terms of the expanded form for CIMs.

Definition 2. [Nodelman et al., 2002] *Amalgamation is an operation which takes two CIMs $Q_{S_1|C_1}$, $Q_{S_2|C_2}$, and forms the new CIM $Q_{S|C}$ where $S = S_1 \cup S_2$ and $C = (C_1 \cup C_2)/S$. Given a fixed ordering $\xi_{S,C}$, we expand $Q_{S_1|C_1}$ and $Q_{S_2|C_2}$ into single matrices over $S \times C$ and then define the amalgamated matrix as the sum $Q_{S|C} = Q_{S_1|C_1} + Q_{S_2|C_2}$.*

Definition 3. [Nodelman, 2007] *The inverse of amalgamation is computed by matrix subtraction.*

3.4 Generative semantics

In this section, we report the semantics of CTBNs by specifying a generative process, a procedure that takes the initial distribution and the description of each local variable and randomly samples a trajectory for the system. The trajectory means a set of instantiations of the variables. The procedure, as Nodelman proposed [Nodelman, 2007] is shown in Algorithm (1); it takes in input an end time and returns a sampled trajectory σ ending at that time. For each variable $X \in \mathbf{X}$, it maintains $x(t)$ - the state of X at time t , and $Time(X)$ - the next potential transition time for X . We will use $\mathbf{u}_X(t)$ to represent the instantiation to parents of X at time t . The initial distribution is defined by the Bayesian network $\mathbf{B}$. For every $X \in \mathbf{X}$ and every $x \in Val(X)$, the parameters of the multinomial distribution, $\theta_{x|\mathbf{u}_X}$ are set such that:

$$\theta_{xx'|\mathbf{u}_X} = \frac{q_{xx'|\mathbf{u}_X}}{\sum_{x'} q_{xx'|\mathbf{u}_X}}, x' \neq x \quad (3.1)$$

Algorithm 1 CTBN-Sample

Require: t_{end} .

- 1: $t \leftarrow 0, \sigma \leftarrow \emptyset$
  - 2: **for** each variable $X \in \mathbf{X}$ **do**
  - 3:   Choose state $x(0)$ according $\theta_{X|paB(X)}^B$.
  - 4: **end for**
  - 5: **loop**
  - 6:   For each variable X such that $Time(X)$ is undefined:
  - 7:   Choose Δt for next X transition from an exponential dist. with parameter $q_{x(t)|uX(t)}$.
  - 8:   Define $Time(X) \leftarrow t + \Delta t$
  - 9:   Let $X = \arg \min_{Y \in \mathbf{X}} [Time(Y)]$
  - 10:   **if** $t = t_{end}$ **then**
  - 11:   **return** σ
  - 12:   **end if**
  - 13:   Update $t \leftarrow Time(X)$
  - 14:   Choose $x(t)$ the next value of X from the multinomial with parameters $\theta_{x(t)|uX(t)}$.
  - 15:   Add $\prec X \leftarrow x(t), t \succ$ to σ .
  - 16: **end loop**
  - 17: Undefine $Time(X)$ and $Time(Y)$ for all variables Y for which $X \in U_Y$.
-

Chapter 4

Inference in Continuous Time Bayesian Networks

In this chapter an overview of the inference methods used in continuous time bayesian networks is offered. Given a CTBN model, composed by the qualitative and quantitative components, we would like to use it to answer queries conditioned on the available observations. Or in many situations, we would like to estimate the parameters of a CTBN model based on some partially observed evidence. All these tasks refer to the so call inference in CTBNs. In particular, the excursus covers the exact inference procedure proposed by Nodelman [Nodelman et al., 2002] and a series of queries to which we can answer. Unfortunately, exact inference has some computational limits, well known in Bayesian Networks and even worst in CTBN. This limitations suggest to use approximate inference which can take computational advantages of the independences encoded in the factored CTBN representation. In the following sections a variant of the Expectation Propagation algorithm is described [Nodelman et al., 2005]. Another way to perform approximate inference in CTBNs is to use sampling procedures. Sampling has several advantages; it is easy to implement and does not require complex numeric computations. It is an anytime algorithm: we can stop at any time and use the obtained samples to compute the answer. The Gibbs Sampling algorithm has been proposed by El-Hay [El-Hay et al., 2008]; together with the Fan's contribution [Fan and Shelton, 2008] they constitute

the state of the art of the sampling procedure for CTBN inference.

4.1 Queries and exact inference over CTBN

As we mentioned in the chapter concerning the CTBN model, the underpinning of this probabilistic framework is based on joint homogeneous Markov processes. Regarding this aspect, it is possible to use a CTBN to answer any query that we can answer using an explicit representation of a Markov process. Evidence for a CTBN is usually a partial trajectory, in which some values or transitions are missing for some variables during some time intervals. There are two common types of observations: *point evidence* and *continuous evidence*. Point evidence represents the observation of the value of some variables at a particular time instant. Continuous evidence provides the behavior of some variables throughout an interval $[t_1, t_2)$ [Fan and Shelton, 2009]. When continuous evidence is available during the inference task, it is possible to avoid to explore a part of the full joint intensity matrix; this aspect is an advantage from the computational point of view.

In [Nodelman et al., 2002] it has been shown that a CTBN $\aleph$ is a factored representation of a homogeneous Markov process described by the joint intensity matrix defined as

$$Q_{\aleph} = \prod_{X \in \mathbf{X}} Q_{X|pa(X)} \quad (4.1)$$

Therefore, the CTBN $\aleph$ can be used to answer any query which can be answered by using an explicit representation of a Markov process. Indeed, given the set of CIMs $Q_{X|pa(X)}$, $X \in \mathbf{X}$ associated with the nodes of the CTBN model $\aleph$ it is always possible to form the joint intensity matrix $Q_{\aleph}$ and then to answer queries just as we do for any homogeneous Markov process. Given the joint intensity matrix $Q_{\aleph}$ and the initial distribution $P_{\aleph}^0$, many questions can be answered about the homogeneous Markov process $\aleph(t)$.

The, distribution over the value of $\aleph(t)$ is given by

$$P_{\aleph}(t) = P_{\aleph}^0 \exp(\mathbf{Q}_{\aleph} t) \quad (4.2)$$

while the joint distribution over any two time points can be computed as follows:

$$P_{\aleph}(s, t) = P_{\aleph}(s) \exp(\mathbf{Q}_{\aleph}(t - s)), \quad t \geq s \quad (4.3)$$

Example 4. Consider the Example 1 of Chapter 3, and assume that from time $t = 0$ to time $t = 1$ the states of the random variables are unknown. Then, from $t = 1$ to $t = 2$ we observe that the random variable X is in state x_1 , from $t = 2$ till $t = 3$ no evidence is available, while from $t = 3$ to $t = 4$ we observe that the random variable Y is in state y_2 . A possible query over the CTBN model, consists of computing the marginal probability for the random variable Z over the entire time interval from $t = 0$ to $t = 4$. The posterior probability, over the time interval $[0, 4]$, for states $X = x_1$, $Y = y_1$ and $Z = z_1$ is plotted in Figure 4.1.

The posterior probability over time in Example 4 has been computed by implementing *amalgamation*, which as described in [Nodelman et al., 2002] and reported in section 3.3. Then the evidence is incorporated into $\mathbf{Q}$ as follows. We reduce the joint intensity matrix $\mathbf{Q}$ to $\mathbf{Q}_i$ for each interval $[t_i, t_{i+1})$ by zeroing out the rows and columns of $\mathbf{Q}$ which represent states that are inconsistent with the evidence. Additionally, let $\mathbf{Q}_{i,j}$ be the matrix $\mathbf{Q}$ with all elements zeroed out except the off-diagonal elements that represent the intensities of transitioning from non-zero rows in $\mathbf{Q}_i$ to non-zero columns in $\mathbf{Q}_j$. If evidence blocks i and j differs only in which variables are observed (no transition is observed between them), then $\mathbf{Q}_{i,j}$ is the identity matrix instead.

$\exp(\mathbf{Q}_i(t_{i+1} - t_i))$ represents the transition matrix for interval $[t_i, t_{i+1})$ and $\mathbf{Q}_{i,i+1}$ corresponds to the transition probability density between two consecutive intervals at time t_{i+1} .

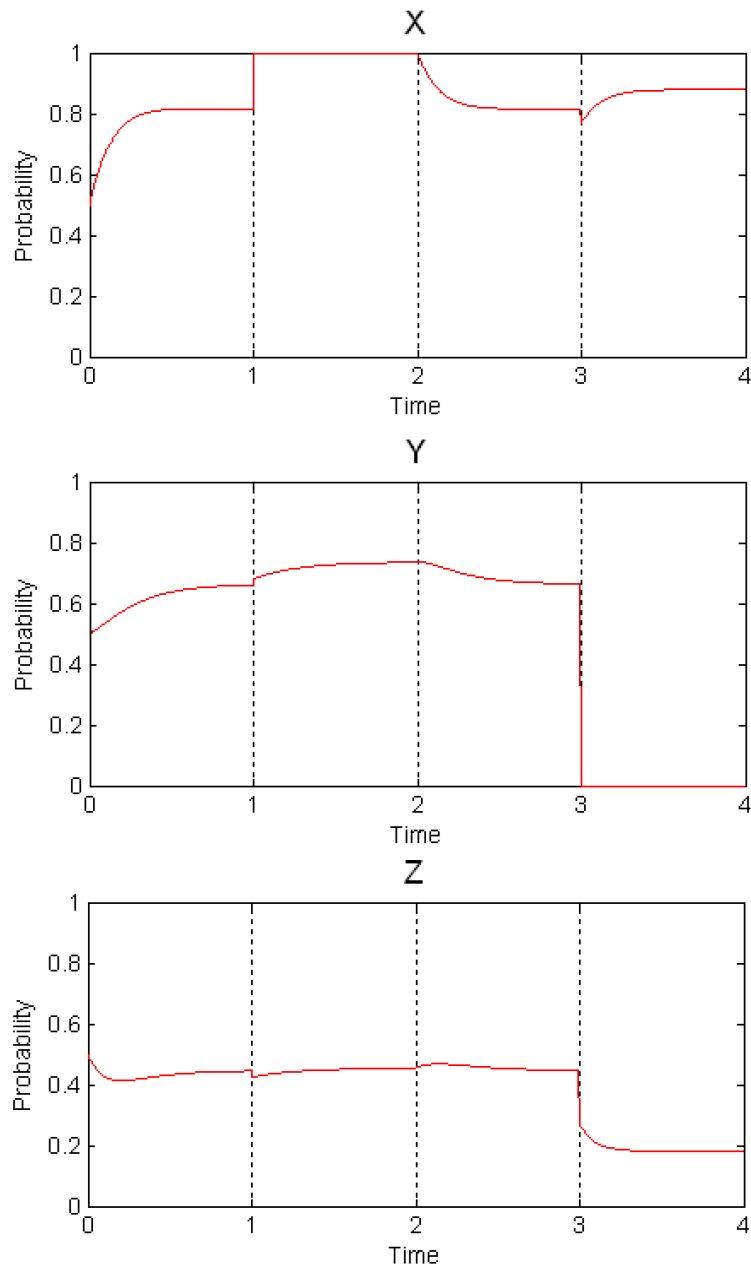


Figure 4.1: Posterior of X , Y , and Z over time for the CTBN of Example 3.1

4.2 Approximate inference: Expectation Propagation

Exact inference in a CTBN requires generating a single joint intensity matrix over the entire state space and calculating the exponential of the matrix Equation (4.2). This step requires a number of parameters which grows exponentially with the size of the network. Hence, exact inference is intractable for large networks. Therefore, approximate inference method is used.

Since a CTBN model uses a graphical representation, inference methods based on message passing can be applied. Nodelman [Nodelman et al., 2005] proposed an algorithm that uses message passing in cluster graphs, of which clique tree algorithms are a special case. In cluster graph algorithms, the nodes of the graph correspond to clusters of variables, and messages between these clusters pass in order to produce an alternative parameterization, in which the marginal distribution of the variables in each cluster can be read directly from the cluster. There are several forms of message passing algorithm. The variant of the expectation propagation (EP) [Minka, 2001] proposed by Nodelman [Nodelman et al., 2005] is based on multiply-marginalize-divide scheme by Lauritzen and Spiegelhalter [Lauritzen and Spiegelhalter, 1988]. Expectation Propagation addresses the problem where messages can be too complex to represent and manipulate by using approximate messages, projecting each message $\delta_{i \rightarrow j}$ from C_i to C_j into a compactly representable space so as to minimize the KL-divergence between $\delta_{i \rightarrow j}$ and its approximation $\hat{\delta}_{i \rightarrow j}$. In a prototypical example [Minka, 2001], the cluster potentials and therefore the sepset marginals are mixtures of Gaussians, which are projected into the space of Gaussian distributions in the message approximation step. For messages in the exponential family, $\operatorname{argmin}_{\delta_{i \rightarrow j}} \mathbf{D}(\delta_{i \rightarrow j} \parallel \hat{\delta}_{i \rightarrow j})$ can be obtained by matching moments of the distribution. EP can be applied to clique trees or to general cluster graphs. In order to improve the quality of approximation the iterative EP algorithm for message propagation has been proposed.

4.2.1 Expectation propagation for segments

We first consider the message propagation algorithm for one segment of our trajectory, with constant continuous evidence [Nodelman et al., 2005]. The first step consists in the construction of the cluster tree for the graph G . This procedure is exactly the same as in Bayesian networks; keep in mind that cycles can be treated in CTBN. Moralization of the graph, connection of all parents of a node with undirected edges, and then linkage all the remaining edges undirected are the actions. If there is a cycle, it simply turns into a loop in the resulting undirected graph. Then it is possible to select a set of clusters C_i . These clusters can be selected in order to produce a clique tree for the graph. Let $A_i \subseteq C_i$ be the set of variables whose factors we associate with cluster C_i . Let N_i be the set of neighboring clusters for C_i and let $S_{i,j}$ be the set of variables in $C_i \cap C_j$. We also compute, for each cluster C_i , the initial distribution $P_{C_i}^0$ using standard Bayesian network inference on the network B . After initialization, the procedure of the algorithm proposed by Nodelman is:

Algorithm 2 Procedure CTBN-Segment EP

Require: $(P^0, T, e, G,)$

- 1: **for** each cluster C_i **do**
- 2: $\pi_i \leftarrow \prod_{X \in A_i} \phi(X, U_X, e)$
- 3: **end for**
- 4: **for** each edge $C_i - C_j$ **do**
- 5: $\mu_{i,j} \leftarrow 1$
- 6: **end for**
- 7: **repeat**
- 8:   Choose $C_i - C_j$
- 9:   Send-message($i, j, P_{C_i}^0, T$)
- 10: **until** convergence

It takes the initial distribution over the clusters P^0 , an amount of time T , and possibly some continuous evidence e which holds for the total time T . $\phi(\cdot, e)$ is used to denote the CIM reduced by continuous evidence e if applicable. The algorithm iteratively selects an edge (i, j) in the cluster graph, and passes a message from C_i to C_j . In clique tree propagation, the edges

---

Algorithm 3 Procedure Send-message

Require: (i, j, P^0, T)

- 1: $\delta_{i \rightarrow j} \leftarrow \text{marg}_{C_i S_{i,j}}^{P^0, T}(\pi_i)$
  - 2: $\pi_j \leftarrow \pi_j \cdot \frac{\delta_{i \rightarrow j}}{\mu_{i,j}}$
  - 3: $\mu_{i,j} \leftarrow \delta_{i \rightarrow j}$
-

have to be selected so as to iteratively perform an upward and downward pass. Convergence occurs when messages cease to affect the potentials which means that neighboring clusters C_i and C_j agree on the approximate marginals over the variables $S_{i,j}$. The basic operations are performed as described in [Nodelman et al., 2005, Nodelman, 2007]. In particular, let $\rho(\cdot)$ be a function taking factors to their CIM parameterization. For the initial potentials, $\rho(\pi_i)$ is computed by adding the intensity matrices $Q_{X|U_X}$ reduced by evidence e for $X \in A_i$. Also, $\rho(1)$ is an intensity matrix of zeros. Factor product is implemented as addition of intensity matrices, and factor division as subtraction, so that $\rho(\pi_j \cdot \frac{\delta_{i \rightarrow j}}{\mu_{i,j}}) = \rho(\pi_j) + \rho(\delta_{i \rightarrow j}) - \rho(\mu_{i,j})$. Marginalization is implemented by computing the expected sufficient statistics, using the evidence e , the time period T , and the initial distribution P^0 , for major details see [Nodelman et al., 2005].

4.3 Stochastic simulation

The advantages of sampling procedures have been explored in order to be implemented as inferential engine for the CTBN. The sampling algorithm generates a trajectory by sampling the transition time and transition state naturally from exponential distribution, which is very easy to implement.

4.3.1 Forward sampling for CTBN

When queries are not conditioned on any observation can be answered by randomly sampling many trajectories and looking at the fraction that match the query. If we have a CTBN $\aleph$ it is possible to generate a set of

---

Algorithm 4 CTBN-Sample

Require: t_{end} .

```

1:  $t \leftarrow 0, \sigma \leftarrow \emptyset$ 
2: for each variable  $X \in \mathbf{X}$  do
3:   Choose state  $x(0)$  according  $\theta_{X|paB(X)}^B$ .
4: end for
5: loop
6:   For each variable  $X$  such that  $Time(X)$  is undefined:
7:   Choose  $\Delta t$  for next  $X$  transition from an exponential dist. with pa-
      rameter  $q_{x(t)|uX(t)}$ .
8:   Define  $Time(X) \leftarrow t + \Delta t$ 
9:   Let  $X = \arg \min_{Y \in \mathbf{X}} [Time(Y)]$ 
10:  if  $t = t_{end}$  then
11:    return  $\sigma$ 
12:  end if
13:  Update  $t \leftarrow Time(X)$ 
14:  Choose  $x(t)$  the next value of  $X$  from the multinomial with parame-
      ters  $\theta_{x(t)|uX(t)}$ .
15:  Add  $\prec X \leftarrow x(t), t \succ$  to  $\sigma$ .
16: end loop
17: Undefine  $Time(X)$  and  $Time(Y)$  for all variables  $Y$  for which  $X \in U_Y$ .
```

---

particles $D = \{\sigma[1], \dots, \sigma[M]\}$ where each particle is a sampled trajectory. D makes possible to estimate the expectation of any function g by computing

$$\hat{E}_{\mathbb{N}}[g] = \frac{1}{M} \sum_{m=1}^M g(\sigma[m]). \quad (4.4)$$

The function $g(\sigma)$ might count the total number of times that X transitions from x_1 to x_2 while its parent U has value u_1 , allowing us to estimate the expected sufficient statistic $M[x_1, x_2|u_1]$ using the same procedure proposed by Nodelman et al. for the generative semantic [Nodelman, 2007]. The forward sampling procedure is presented by Fan and Shelton [Fan and Shelton, 2008] and is shown in Algorithm (4).

For each variable $X \in \mathbf{X}$, it maintains $x(t)$ and $Time(X)$. The transitions are added one at a time, advancing t to the next earliest variable transition. When a variable X (or one of its parents) undergoes a transition,

$Time(X)$ is resampled from the new exponential waiting time distribution. $u_X(t)$ is used to represent the instantiation of the parents of X at time t . If we want to obtain a conditional probability of a query given evidence, the situation is more complicated. We might try to use rejection sampling: forward sample to generate possible trajectories, and then delete which ones that are inconsistent with the evidence available. The remaining trajectories are sampled from the posterior distribution given the evidence, and can be used to estimate probabilities as in Equation (4.4). However, this approach is impractical in any setting involving an observation of a continuous quantity such as time. Suppose it is possible to observe that X transitions from x_1 to x_2 at time t . The probability of sampling a trajectory in which that transition occurs at precisely that time is zero. Thus, if we have evidence about transitions, with probability 1, none of the sampled trajectories will be relevant [Fan and Shelton, 2008].

4.3.2 Gibbs Sampling for CTBN

Fan and Shelton [Fan et al., 2010] introduced a procedure that employs importance sampling and particle filtering to sample trajectories from the continuous time Bayesian network. Such a stochastic sampling procedure has anytime properties as collecting more samples leads to more accurate approximation. However when the evidence is mostly at the end of the relevant time interval, and is of low probability, the procedure requires many samples. El Hay et al. [El-Hay et al., 2008] proposed a new stochastic sampling procedure for factored Continuous Time Markov Processes. The goal is to sample random system trajectories from the posterior distribution. Once multiple independent samples are available from this distribution it is possible to approximate the answer to queries about the posterior using the empirical distribution of the samples. The challenge is to sample from the posterior. Sampling given evidence is not so trivial, as evidence modifies the posterior probability of earlier time points. The Markov Chain Monte Carlo (MCMC) procedures proposed by El Hay circumvent this problem by sampling a trajectory that will eventually be governed by the desired posterior distribution. A Gibbs sampling procedure for factored CTMPs

have been developed. This procedure is initialized by setting an arbitrary trajectory which is consistent with the evidence. It then alternates between randomly picking a component X_i and sampling a trajectory from the distribution of X_i conditioned on the trajectories of the other components and the evidence. The algorithm proposed by El Hay samples a trajectory for a single component from a process that is conditioned on trajectories of the other components. While such a process is Markovian, it is not homogeneous as its dynamics depends on trajectories of its Markov Blanket as well as on past and present evidence. It is possible to perform exact sampling by utilizing this Markovian property, and the cost of this procedure is determined by the complexity of the current trajectories and the sampled one, and not by a predefined resolution parameter. This implies that the computational time of the stochastic procedure adapts to the complexity of the sampled object [El-Hay et al., 2008]. While a single trajectory is defined over infinite time points, it involves only a finite number of transitions in a finite interval. Therefore, instead of sampling states at different time points, the proposed procedure samples a finite sequence of transitions. The procedure starts sampling the first transition time. It then samples the new state the transition leads to. As this new sample point statistically separates the remaining interval from the past, we are back with the initial problem yet with a shorter interval. These steps are repeated until the entire trajectory is sampled; it terminates once the next transition time is past the end horizon of the interval. Before sampling the first transition time, it is necessary to define the conditional cumulative distribution function $F(t)$ that the node X of the CTBN stays in the initial state for a time less than t :

$$F(t) = 1 - Pr(X^{(0,t]} = x^{(0)} | x^{(0)}, x^{(T)}, y^{[0,T]}) \quad (4.5)$$

The Markov property of the conditional process allows to decompose the probability that X remains in its initial state until time t . The probability of Y 's trajectory and of X remaining in its initial state until time t is given by:

$$p^{past}(t) = Pr(X^{(0,t]} = x^{(0)}, y^{(0,t]} | x^{(0)}, y^{(0)}), \quad (4.6)$$

and the probability of future observations given the state of (X_t, Y_t) is:

$$p_x^{future}(t) = Pr(x^{(T)}, y^{(t,T)} | X^{(t)} = x, y^{(t)}). \quad (4.7)$$

Consequently, it is possible to define the probability of X is in state $x^{(0)}$ until time t :

$$Pr(X^{(0,t]} = x^{(0)} | x^{(0)}, x^{(T)}, y^{[0,T]}) = \frac{p^{past}(t) \cdot p_{x^0}^{future}(t)}{p_{x^0}^{future}(0)} \quad (4.8)$$

For details concerning the computation of $p^{past}(t)$ and $p_{x^0}^{future}(t)$ see [El-Hay et al., 2008].

Chapter 5

CTBN toolbox

In this chapter a software toolbox for inference in Continuous Time Bayesian Networks is proposed. The system implements exact and approximate inference. Exact inference requires full amalgamation of conditional intensity matrices, described in the previous chapters. Therefore this type of inference quickly becomes intractable and has been implemented for comparison purposes only. Approximate inference is implemented through the expectation propagation algorithm. The software environment is implemented in Matlab to exploit its numerical efficiency on matrix operations which are the core of exact and expectation propagation algorithms. Point evidence, usually available also in Dynamic Bayesian Networks, and interval evidence, which is one of the main modeling advantages of Continuous Time Bayesian Networks, are available. The output consists of posterior probability values through time for each non observed node.

5.1 The Matlab CTBNs Environment

The toolbox has been implemented in Matlab. Each CTBN model can be specified by two files namely, *model file* and *evidence file*. These files are the input to the Matlab CTBNs environment which offers two algorithms for forward inference: the CTBN full amalgamation and the Expectation Propagation (EP) algorithm. Both algorithms allow to cope with point and interval evidence and output a Matlab figure associated with each node of the

CTBN model. Each figure depicts the temporal evolution of the probability values associated with each state of the considered node. Furthermore, the algorithms save these results to a structured and highly readable excel '.csv' file. To better describe the main components of the software toolbox we use the drug effect network introduced by Nodelman et al. [Nodelman et al., 2002]. This CTBN model consists of the following nodes; *Hungry*, *Eating*, *Full-Stomach*, *Uptake*, *Concentration*, *Barometer*, *Joint-Pain* and *Drowsy*. All nodes are binary except *Barometer* which is ternary. The qualitative component of the drug effect network, i.e. the graph, is depicted in Figure 5.1. It is worthwhile to mention that in this CTBN model the nodes *Full-Stomach*, *Hungry* and *Eating* form a cycle.

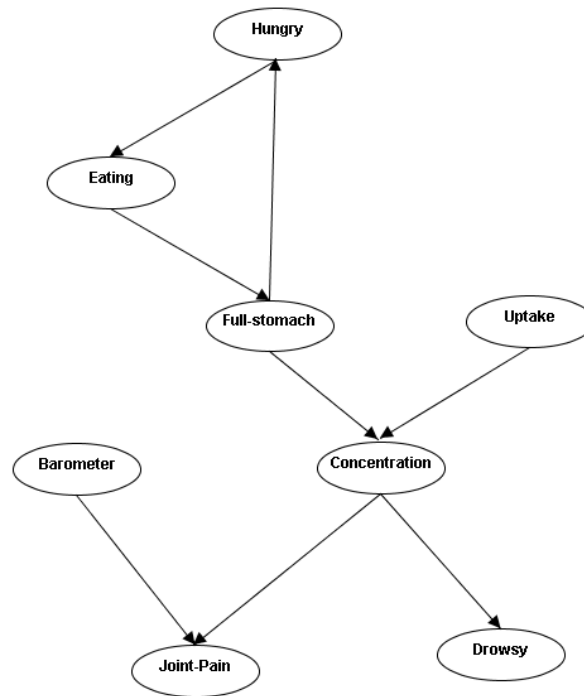


Figure 5.1: Drug effect network

5.1.1 Model file

A particular syntax has been proposed both for the model and the evidence file. The structure of a CTBN model, qualitative and quantitative components, is specified through an input file structured into three parts. The first part allows to define the number of nodes of the CTBN and for each node to specify its name and cardinality (Figure 5.2).

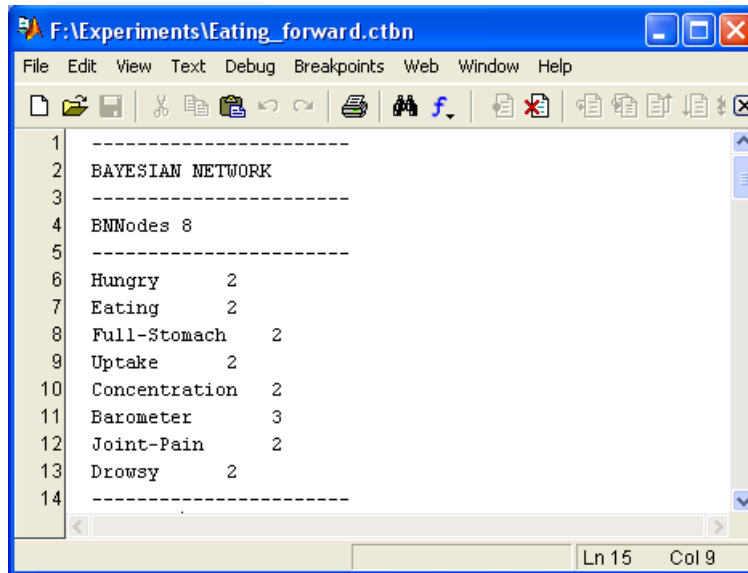


Figure 5.2: CTBN model file: node's name and their cardinality for the CTBN shown in Figure 5.1

The second part is associated with the static BN model. It allows to describe the directed acyclic graph as well as the conditional probability table associated with each node of the BN (Figure 5.3(a)). It is worthwhile to mention that for the drug effect network each conditional probability table is assumed to implement a binomial or multinomial with uniform probability for each state of the considered node.

Furthermore, the conditional probability table is structured in such a way that rows are associated with states of the considered node whereas columns are associated with joint configurations of its parent nodes. The third part is associated with the possibly cyclic graph and the CIMs associ-

ated with each node of the CTBN. Given a node of the CTBN, its name is reported followed by the list of its parents which stops when a '0' is found. The CIMs associated with each node are specified as a set of rows. Each row is associated with a joint configuration of the node's parents and is sorted in such a way that subsequent CIM rows are concatenated on a single line (Figure 5.3(b)).

In particular the node *Hungry* is binary and has the parent *Full-Stomach* which is binary also. Therefore, the node *Hungry* will be described with two CIMs, one associated with the configuration *Full-Stomach=1* and the other associated with the configuration *Full-Stomach=2*. The CIMs are as follows:

$$\mathbf{Q}_{Hungry \mid Full-stomach=1} = \begin{bmatrix} -1 & 1 \\ 10 & -10 \end{bmatrix} \quad \mathbf{Q}_{Hungry \mid Full-stomach=2} = \begin{bmatrix} -20 & 20 \\ 5 & -5 \end{bmatrix}$$

and can be easily identified at the bottom of Figure 5.3(b).

5.1.2 Evidence file

The inference process is performed for a given time horizon and depending on the available point and/or continuous evidence. These information are specified through the evidence file. Let us assume that for the *drug effect network* the following stream of evidence is available: at time $t = 0$ we know that the patient is hungry (*Hungry=1*), the concentration (*Concentration=2*) is high and the barometer is low (*Barometer= 1*), then till time $t = 2$ we do not have any evidence while from time $t = 2$ to time $t = 3$ we observe that the patient is eating (*Eating=1*). The corresponding evidence file is depicted in Figure 5.4. The evidence stream:

' . [0.0] (3) *Hungry* = 1 *Concentration* = 2 *Barometer* = 1'

indicates that the evidence stream is a point evidence (.) and is related to time 0 ([0.0]), where three (3) nodes are evidenced. The evidenced nodes are *Hungry* in state 1, *Concentration* in state 2 and *Barometer* in state 1. Each evidence stream starts with one of the following symbols: '.', '-' and '?'. While '.' is associated with point evidence and '-' with interval evidence, '?' is associated with no evidence. The symbols representing the evidence are

```

F:\Experiments\Eating_forward.ctbn
File Edit View Text Debug Breakpoints Web Window Help
-----
14
15 Hungry 0
16 Eating Hungry 0
17 Full-Stomach Hungry Eating 0
18 Uptake 0
19 Concentration Full-Stomach Uptake 0
20 Barometer 0
21 Joint-Pain Concentration Barometer 0
22 Drowsy Concentration 0
23 -----
24 Hungry
25 0.50 0.50
26 -----
27 Eating
28 0.50 0.50
29 0.50 0.50
30 -----
31 Full-Stomach
32 0.50 0.50 0.50 0.50
33 0.50 0.50 0.50 0.50
34 -----
Ln 15 Col 9

```

(a) Structure of the BN's DAG and part of the CPTs

```

F:\Experiments\Eating_forward.ctbn
File Edit View Text Debug Breakpoints Web Window Help
-----
53
54 DIRECTED GRAPH
55 -----
56 Hungry Full-Stomach 0
57 Eating Hungry 0
58 Full-Stomach Eating 0
59 Uptake 0
60 Concentration Full-Stomach Uptake 0
61 Barometer 0
62 Joint-Pain Concentration Barometer 0
63 Drowsy Concentration 0
64 -----
65
66 CIMS
67 -----
68 Hungry
69 -1.00 1.00 10.00 -10.00
70 -20.00 20.00 5.00 -5.00
71 -----
72 Eating
73 -0.01 0.01 10.00 -10.00
74 -2.00 2.00 0.01 -0.01
75 -----
Ln 15 Col 9

```

(b) Structure of the CTBN's graph and CIMS

Figure 5.3: CTBN model file.

followed by a bracketed number which specifies the number of evidenced nodes, whose name and evidence state are listed on the right. The evidence file terminates with the '*' symbol.

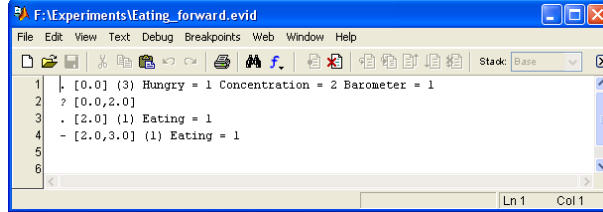


Figure 5.4: CTBN evidence file.

5.1.3 Inference engines

Two inference engines are available: CTBN full amalgamation and Expectation Propagation. CTBN full amalgamation is based on CIM amalgamation [Nodelman et al., 2002]. This operation combines two CIMs to produce a single, larger CIM. The CTBN full amalgamation algorithm consists of amalgamating all the CIMs of the CTBN to get a single intensity matrix which encodes the probability distribution over the dynamics of the analyzed system. However, exact inference is intractable even for small CTBN models and thus we have to consider approximate inference algorithms. Therefore, the CTBNs Matlab software environment makes available an optimized implementation of the EP algorithm, through the integration of software code previously designed and developed at the MAD laboratory (Models and Algorithm for Data and Text Mining laboratory of the University of Milano Bicocca). Its main software components are the following:

- *CTBN-initialize-ctbn-model*: takes the CTBN model, qualitative and quantitative components, and initializes the EP algorithm. It computes the CTBN clique tree, amalgamates CIMs to compute clique potentials and sets the initial message associated with edges to one. Then it invokes the *CTBN-initialize-data-structure* software component;

- *CTBN-initialize-data-structure*: initializes all the required EP data structures as well as all data structures used to record output informations. It exploits the powerful cell array Matlab data structure which provides a storage mechanism for different data types;
- *CTBN-EP-Filter*: implements filtering - forward message passing. It uses the joint probability distribution over the CTBN at time zero, a list of times, where each time is associated with point and interval evidence. For each time interval of constant evidence it invokes the CTBN-EP-Segment software component;
- *CTBN-EP-Segment*: takes the initial distributions over the cliques of the CTBN, an amount of time and some continuous evidence (interval evidence) lasting for the entire amount of time. It is an optimized version of the EP for segments algorithm described in [Nodelman et al., 2005]. It invokes several times, depending on the required precision, the CTBN-EP-temporal-propagation software component;
- *CTBN-EP-temporal-propagation*: takes the CTBN clique tree, the clique potentials, the available interval evidence and performs approximate message passing to compute sufficient statistics, posterior CIMs for each clique and posterior probability for each node;
- *CTBN-EP-static-calibration*: takes the CTBN clique tree, the clique potentials, and calibrates clique and separating set marginals to obtain a coherent distribution. Indeed, even if the EP computation converges two clusters only agree on the projected marginals over their separating set;
- *CTBN-EP-static-propagation*: takes the CTBN clique tree, the clique potentials, the available point evidence and performs message-passing to compute posterior for each node.

The EP engine allows to decide the quality of the approximation by allowing to refine or coarse the choice of segments. This feature is particularly important to provide the necessary flexibility when dealing with variables that change rapidly even in the case when evidence remains constant.

5.1.4 Output interfaces

Inference engines output probability values associated with each node of the CTBN model for each considered time point. These information are made available through the following interfaces:

- Matlab figure: shows how probability values for each state of a node changes over time. Figure 5.5 shows the output for the Concentration node;
- Excel "csv" file: contains, in a structured format, the informations concerning the probability values associated with each node. Sufficient statistics are reported thus increasing the amount of information available to analyze the output of the inferential process.

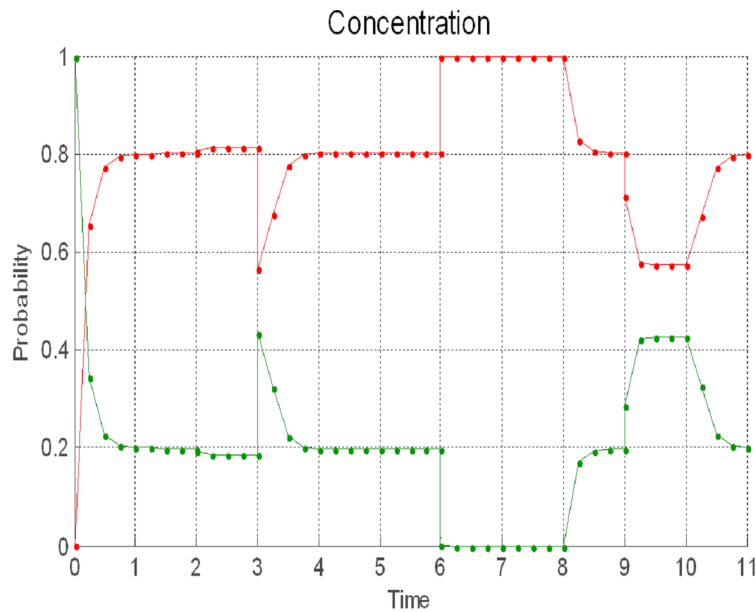


Figure 5.5: Concentration posterior probability of both states of the node.

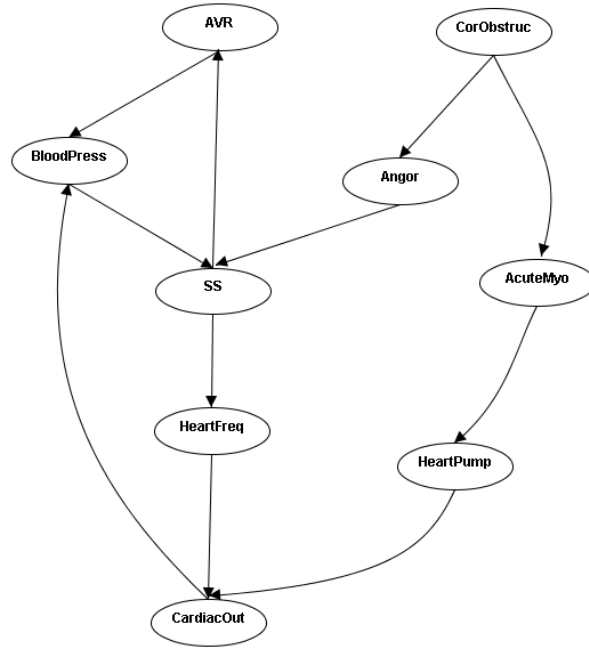


Figure 5.6: Toy heart disease network

5.1.5 Performance analysis

This section summarizes the results of a performance analysis task concerning the trade-off between computational time and precision of the EP implementation. To this extent the drug effect network and a toy heart disease network have been used.

The heart disease network considered in this section is a toy model to make inference of the acute myocardial infarction. It consists of nine nodes, five binary and four ternary. The directed graph is depicted in Figure 5.6 where the following two cycles can be easily identified: *SS-AVR-BloodPres* and *BloodPres-SS-HeartFreq-CardiacOut*.

These networks have been queried through an empty scenario, i.e. a scenario without any evidence and by using 4 scenarios with different settings of point and interval evidence. In Table 5.1 the results of this experiments are summarized. The number of states (States) represents the number of elements of the fully amalgamated CTBN model, i.e. the number of ele-

				Time (seconds)				
				Total		Per Step		MaxAD
Model	States	Sc.	St.	Exa	App	Exa	App	
Drug	147,456	0	2	8.47	0.61	4.23	0.31	0.0005
		1	7	9.77	1.09	1.40	0.16	0.0128
		2	8	9.09	1.15	1.14	0.14	0.0317
		3	12	12.37	1.41	1.03	0.12	0.0557
		4	13	8.99	1.49	0.69	0.12	0.0558
Heart	6,718,464	0	2	1,635.01	0.90	817.50	0.45	0.0137
		1	2	1,629.19	0.89	814.59	0.45	0.0027
		2	7	1,819.01	2.95	259.85	0.42	0.0483
		3	4	1,631.05	1.64	407.75	0.41	0.0035
		4	2	1,527.66	0.99	763.83	0.48	0.0003

Table 5.1: Table of the performance analysis

ments of the resulting intensity matrix. Scenario (Sc.) is associated with scenarios used to query the CTBN models. Scenario 0 is associated with no evidence while scenarios from 1 to 4 are associated with different streams of point and interval evidence. Steps (St.) represents the number of computing steps of the scenario. Total and Per Step times are reported for both the Exact (Exa) and Approximate (App) inference engines. Furthermore, the maximum absolute difference between the true probability and the EP approximated value (MaxAD) is reported.

This chapter describes the Matlab toolbox designed and implemented for the inference task in CTBNs. Two inference engines modules are available: exact inference performed through full amalgamation and Expectation Propagation on CTBN clique tree using the procedure described in Chapter 4. The performed numerical experiments on the drug effect CTBN and on the toy heart disease network together with other experiments (results are not shown) allow to conclude that the precision of the proposed implementation of the Expectation Propagation algorithm is satisfactory while the computational time could be further reduced.

Chapter 6

A CTBN model for Cardiogenic Heart Failure

This chapter describes how continuous time Bayesian networks can be used to diagnose the cause of cardiogenic heart failure and to anticipate its likely evolution. The proposed model overcomes the strong modeling and computational limitations of dynamic Bayesian networks. It consists of both unobservable physiological variables, and clinically and instrumentally observable events which might support diagnosis like myocardial infarction and the future occurrence of shock. Three case studies related to cardiogenic heart failure are presented. They include point evidence, usually available also in dynamic Bayesian networks, and interval evidence, one of the main modeling advantages of continuous time Bayesian networks. The model predicts the occurrence of complicating diseases and the persistence of heart failure according to variations of the evidence gathered from the patient. Predictions are shown to be consistent with current pathophysiological medical understanding of clinical pictures [Gatti et al., 2011].

6.1 Introduction

Recent technological developments in the field of Information and Communication Technology (ICT) offered an extremely important opportunity

to operational health care management [ICTBIO, 2008]. Because of that, Decision Support Systems (DSSs) are becoming increasingly attractive for physicians, as they can offer great benefits without necessarily dare to replace human judgment [Lipscombe, 1989, Sim et al., 2001, Garg et al., 2005]. However, medical decisions are based on potentially uncertain information while the contribution of DSSs in health care has been far-reaching and still evolving as evidenced by the large number of references that appear in PUBMED, a widely used health care search engine. Increasing health care costs make it imperative for hospitals and physicians to make optimal decisions to improve the quality and the efficiency of health care delivery. Recent advances in DSSs for health care [Burstein and Holsapple, 2008], provide a prominent and growing role of DSSs in improving clinical as well as administrative decision making.

Bayesian networks [Pearl, 1988] have become a popular representation in AI for encoding uncertain knowledge [Hulst, 2006, Rose et al., 2005]. Several examples exist where they were used for reasoning under uncertainty by medical systems [Xiang et al., 1993]. TakeHeart II, a DSS for assessing heart attack risk [Hope et al., 2007], QMR-DT [Middleton et al., 1991] for internal medicine, Pathfinder [Heckerman et al., 1992] and Intelipath [Nathwani et al., 1990] in pathology, Qualicon [Xiang et al., 1992] in nerve conduction studies, Myosys [Vila et al., 1985] devoted to diagnose mono and poly-neuropathies, Myolog [Gallardo et al., 1987] for diagnosing plexus and root lesions, Electrodiagnostic Assistant [Jamieson, 1990] devoted to diagnose entrapment neuropathies, plexopathies and radiculopathies, Neurop [Besnard et al., 2002] oriented toward neuropathy diagnosis, Kandid [Fuglsang-Frederiksen et al., 1989] and Munin [Andreassen et al., 1989] which allow clinicians to diagnose a complete range of neuromuscular disorders. However, in many diagnostic problems the time duration of events, i.e. patient conditions, is a central element to quickly reach the correct diagnosis. In such cases Bayesian networks have been replaced by their natural extension, i.e. dynamic Bayesian networks [Dean and Kanazawa, 1989], able to represent the evolution over time of the analyzed system. However, dynamic Bayesian networks themselves suffer from expressiveness and computational limitations.

In this chapter a continuous time Bayesian network model is described for cardiogenic heart failure diagnosis and some prognosis scenarios have been developed. Acute cardiogenic heart failure remains one of the most common, costly, disabling and deadly medical conditions encountered by a wide range of physicians and surgeons in both primary and secondary care. The understanding of its epidemiology, pathophysiology, diagnosis, and, especially, treatment has advanced greatly during the past 20 years and continues to develop rapidly. Between 1% and 2% of the adult population have heart failure, although it mainly affects people over 65 years old. The numbers are rapidly increasing because of the aging population. The proposed continuous time Bayesian network overcomes the strong modeling and computational limitations of dynamic Bayesian networks. It consists of physiological and clinical variables, some of which are observable while some others are not. The unobservable variables are the focus of diagnostic judgment like myocardial infarction and cardiac pump impairment, as well as prognosis task related to the occurrence of shock and heart failure persistence. Three scenarios showing how the developed model can be used for early analysis of patients are reported. The described scenarios include point evidence, usually available also through dynamic Bayesian networks, and interval evidence, which is one of the main modeling advantages of continuous time Bayesian networks over dynamic Bayesian networks. The continuous time Bayesian network model accurately represents the cardiovascular system and all mechanisms which have been judged to be relevant to myocardial infarction modeling.

6.2 The Cardiogenic Heart Failure

There is no universally agreed definition of heart failure. Heart failure is a syndrome which develops as a consequence of cardiac disease, and is recognized clinically by a constellation of symptoms and signs produced by complex circulatory and neurohormonal responses to cardiac dysfunction. The syndrome may occur as the end-result of damage caused by a number of disease processes e.g. coronary artery disease, hypertension, valvular de-

fects, alcohol misuse or viral infection. The presence and severity of heart failure can be assessed by questionnaires, physical and radiographic examination, and by measures of ventricular performance and exercise capacity. All these methods, however, have major limitations when used independently. Most diagnostic difficulties arise in individual cases where the syndrome is mild.

6.2.1 Heart failure in the medical domain

Heart failure is a disorder in which the heart pumps blood inadequately. Because heart pumps oxygenated blood into the arterial vessels, while taking not oxygenated blood from the veins, the consequences of heart failure are twofold. On one side, it leads to a reduced blood flow with a lower delivered oxygen into the peripheral tissues, which in turn induces a reduced exercise capacity level and fatigue, or even an irreversible condition known as shock, in which cells become unable to meet their metabolic functional needs. On the other side, heart failure induces congestion of blood both in the veins and lungs, leading to shortness of breath and organs' enlargement.

In order to understand what heart failure means in physiological terms, we should refer to the so called *circulation theory*, introduced for the first time in *Exercitatio Anatomica de Motu Cordis et Sanguinis in Animalibus* by William Harvey in 1628. The right side of the heart pumps blood from the veins into the lungs. Otherwise, the left side pumps blood from the lungs out through the arteries to the rest of the body. Blood goes out when the heart muscle contracts (systole) and comes in when the heart muscle relaxes (diastole). When blood causes congestion in those tissues, heart failure is said to be *congestive*. Accumulation of blood coming into the left side of the heart causes congestion in the lungs, making breathing difficult.

Accumulation of blood coming into the right side of the heart causes congestion and fluid accumulation in other parts of the body, such as the liver and the legs. Indeed, where fluid accumulates depends on the amount of excess fluid and the effects of gravity. If a person is standing, fluid accumulates in the legs and feet. The kidneys filter fluid and waste products

from the blood into the urine, but when the heart cannot pump adequately, the kidneys malfunction and cannot remove excess fluid from the blood. As a result, the amount of fluid in the bloodstream increases, and the workload of the failing heart increases, creating a vicious circle. Thus, heart failure becomes even worse.

Heart failure can be caused by factors originating within the heart or from external factors that place excessive demands upon the heart. When causes are intrinsic, the condition is labeled *cardiogenic heart failure*. The most important cause of cardiogenic heart failure is coronary artery disease. This disorder reduces the oxygen delivery to the myocardium and, as a consequence, the strength of the myocardial pump. Another common cause of heart failure is acute myocardial infarction, often the culmination of a coronary artery disease.

Symptoms of heart failure may arise suddenly, especially if the cause is a heart attack, or they develop gradually over days to months or years. Clinicians usually suspect heart failure on the basis of symptoms alone and how long they last. The diagnosis is supported by the results of a physical examination, including a weak, often rapid pulse, reduced blood pressure and fluid accumulation in the lungs. Several markers can be detected with the aim to diagnose, e.g. a chest X-ray can show an enlarged heart and fluid accumulation in the lungs while the presence of cardiac enzymes proves the occurrence of an acute myocardial infarction.

6.2.2 Time granularity

If we want to represent in a computer system the temporal context associated with real-world facts and events and we want to reason with them, we somehow have to devise a formal characterization of time granularities [Bettini et al., 2000]. The concept of granularity [Hobbs, 1985] is ubiquitous to any temporal system and decisions taken at such level will affect the functioning of the system. Basically dealing with granularity is about providing ways to safely convert temporal references provided in different units, e.g. from hours to minutes or from minutes to seconds. Using different time units is regular practice in medicine. Either in management,

diagnosis, prognosis or treatment, there is always a need to relate processes that develop at a given granularity to those that may happen at a different one. For example a visualization system in a system may allow in a hospital to track the different events associated with a patient during a certain period of time. If the patient has been for, lets say, a month it is probable the patient went through different stages since first diagnosed until the final stages of her/his stay. Views of his evolution throughout one month may focus on the most relevant aspects, like diagnosis, prescription and evolution. A view focused on the activities and significant events of a particular day may focused instead on the patients' response to specific drugs and the results on specific analysis. On a different kind of application for the concept, sometimes symptoms are identified if a given pattern has been occurring each day for a week or so on. Other times an occurring pattern that was present during weeks can be associated to a season, e.g. when diagnosing hay fever. As it can be seen, the problem of handling granularity flexibly and effectively in medicine is quite difficult and challenging. Although granularity is concerned with decisions taken at the temporal structure level, it will also be reflected in the reasoning mechanisms. Hence, a system allowing different level of granularities must supplement that with an accordingly granularity-sensitive inferential system. Since computer system are very fast in making calculations, we may think that a single time granularity, the finest possible, can be used to represent and manipulate time and indeed this approach has been taken in many applications. If we can capture and represent this semantics into the probabilistic framework we can perform effective temporal reasoning and make inference in order to identify the occurrence of events of interest. In Dynamic Bayesian Networks the granularity needs to be selected by the user, in advance, a design choice which is far from obvious. Consider as an example the simple DBN composed by two nodes A and B, A is parent of B; the first is binary and the second ternary. An example of conditional probability table (CPT) for the node B is proposed in Table 6.1

The quantitative counterpart in a Continuous Time Bayesian Network is given by the Conditional Intensity Matrix. As we mentioned before, the intensity taxes of an intensity matrix gives the instantaneous behavior of

CPT	b_1	b_2	b_3
a_1b_1	0.6500	0.2000	0.1500
a_1b_2	0.3000	0.5800	0.1200
a_1b_3	0.2100	0.5500	0.2400
a_2b_1	0.2400	0.7400	0.0200
a_2b_2	0.6000	0.3500	0.0500
a_2b_3	0.6000	0.3000	0.1000

Table 6.1: An example of a Conditional Probability Table in a DBN.

the Markov process which is the underpinning of the CTBN. The specification of the transition model induces a probability distribution over the trajectories. Suppose that the quantities of the CIM of a variable in a CTBN are exponentially distributed, then:

$$\begin{aligned}
 \lim_{\Delta t \rightarrow 0} P(X(t + \Delta t)|X(t)) &= \lim_{\Delta t \rightarrow 0} \exp(Q_x \Delta t) \\
 &= \lim_{\Delta t \rightarrow 0} I + Q_x \Delta t + O(\Delta t^2)
 \end{aligned} \tag{6.1}$$

a contribute to this thesis in order to complete the quantities in a CIM is given by the formula proposed above:

$$q_{ii} = \frac{\ln P(\cdot)}{\Delta t} \tag{6.2}$$

with the formula in (6.2) the intensities taxes to remain in a particular state i can be determined, the q_{ij} quantities of the CIM are calculated using the normalization after excluding the probability of staying in a determined state.

Considering a $\Delta t = 3$ minutes, the corresponding CIM for the node B is depicted in the Table 6.2

For example, the $P(b_1|a_1b_1) = 0.6500$ and using the formula (6.2) $q_{b_{11}|a_1} = -0.1436$. In our case, the elicitation of the CPT for a DBN has been proposed by domain experts.

CIM	b_{11}	b_{12}	b_{13}	b_{21}	b_{22}	b_{23}	b_{31}	b_{32}	b_{33}
a_1	-0.1436	0.0821	0.0615	0.1297	-0.1816	0.0519	0.1314	0.3442	-0.4757
a_2	-0.4757	0.4632	0.0125	0.3230	-0.3499	0.0270	0.5117	0.2558	-0.7675

Table 6.2: An example of a Conditional Intensity Matrix for the node B in a simple CTBN.

6.2.3 The CTBN model for cardiogenic heart failure

The qualitative component of the CTBN model, i.e. the directed cyclic graph modeling the cardiovascular system, is shown in Figure 6.1 while the quantitative component, i.e. the set of CIMs, can be downloaded at the following URL: www.mad.disco.unimib.it/CTBNProject.

The qualitative and quantitative components have been devised on the basis of knowledge elicitation from physicians. The CIMs, have been obtained from a DBN model of the cardiovascular system for a time granularity of 10 seconds, a time-step short enough to capture interesting dynamics. Moreover, this choice of the time-step allows us to neglect the periodical changes of some physiological variables, thus removing the need to split them into their systolic and diastolic counterparts. The DBNs graphical structure provides inferential operators with instructions to reduce the joint probability of all the variables into a product of local probability distributions. Therefore, the computational efficiency is increased by exploiting the Bayes rule which is applied over tables of conditional probabilities. Conditional probabilities can be estimated from data, as long as data on hand provide enough evidence of most of the contemplated variables' values. When an adequate sample is not available, subjective probabilities can be elicited from one or more experts, mirroring their knowledge in the domain of interest [Twardy et al., 2006]. A systematic approach to elicitation and, above all, the ability to reduce the local probability distributions to functions based on a less number of parameters, can greatly increase the quality of the elicited probabilities [Mascherini and Stefanini, 2010]. In the proposed CTBN model, the CIM parameters are not conditional probabilities, like in DBNs. However, the latter were thought to be easier parameters to elicit for a medical expert. Thus, for each node, a physician was asked to

consider the probability of its changes, conditionally on the set of its parent's values within a fixed time-step equal to 10 seconds. Simple operations allow to turn these probabilities into the corresponding CIMs parameter values. To elicit conditional probabilities, the distribution probabilities over all the parents' combinations for each node are parameterized in terms of functions like the Noisy-Or-Gate or Gaussian multivariate models. More complex models are adopted when assumptions behind the correctness of those functions do not hold, like the lack of interactions or additivity of continuous effects for the Noisy-Or-Gate or Gaussian multivariate models respectively. The model in Figure 6.1 allows us to make inference for all the variables based on a limited set of observations. Some of them are easy to be collected, like heart frequency (**HF**) and mean arterial blood pressure (**BP**). Others result from the application of either simple diagnostic procedures, like a pleural effusion (**PleuEff**) by means of a chest x-rays or serum cardiac enzymes (**CardEnz**) by means of blood-gas analysis.

The strength of the heart in pumping blood into the vascular system is represented by the node **Pump**. With the heart frequency (**HF**) variable, the cardiac pump influences both the left and the right cardiac output (**LCO** and **RCO**), as well as the left and right cardiac input (**LCI** and **RCI**). However, the amount of blood coming out from the ventricles is constrained by the availability of blood arriving in the cardiac chamber. Thus, **LCO** depends on the blood pressure within the tract between the pulmonary capillaries and the left ventricle (**PctoLVcirc**), likewise **RCO** depends on the blood pressure within the tract between the capillaries and the right ventricle (**CBRVcirc**). The amount of blood being input into the left and right heart does influence the pressure within two tracts, respectively the vessels between the pulmonary capillaries and the left ventricle (**PctoLVcirc**) and the vessels between the capillaries and the right ventricle (**CBRVcirc**). Two nodes have been added to represent the amount of fluid exchanged with the external environment. The first is labeled blood volume (**BV**), being affected by the balance between the water intake (**WI**) and the urinary output (**UO**). As such, it is supposed to influence the pressure within the systemic venous tract (**CBRVcirc**). The second is labeled urinary output (**UO**), which in turn depends on the blood pressure occurring in the systemic arterial

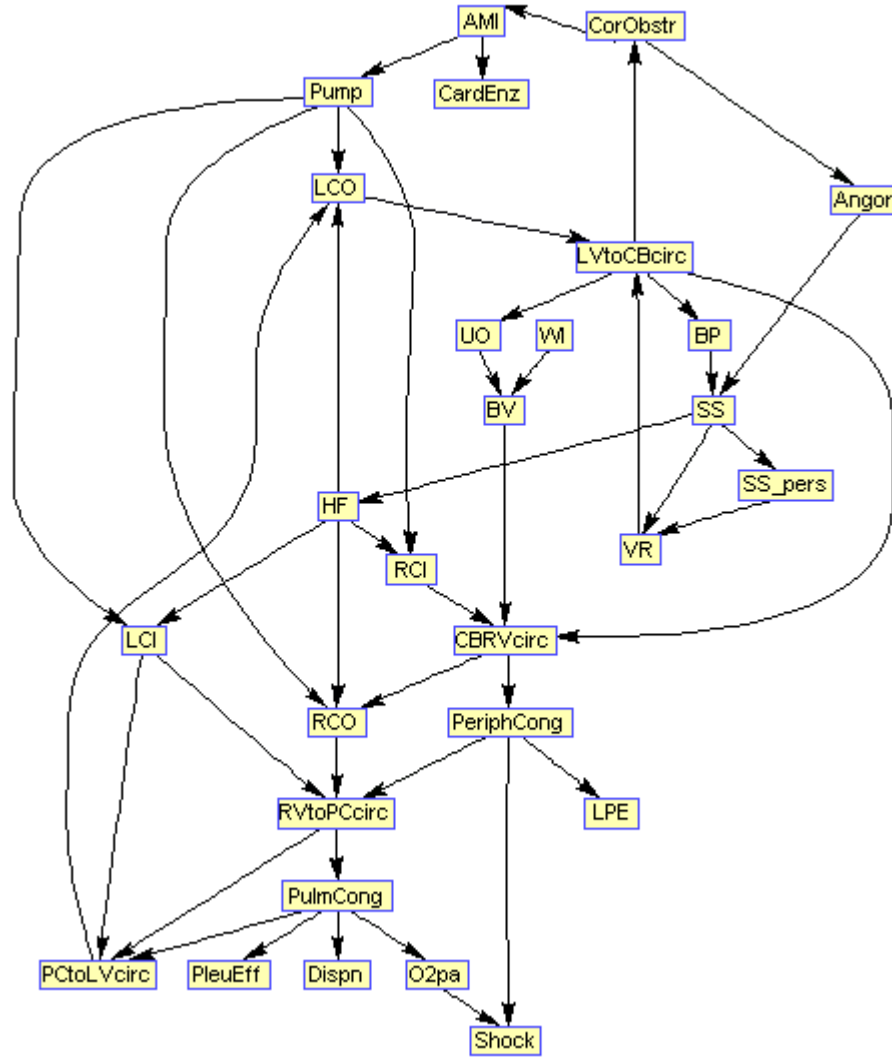


Figure 6.1: CTBN model for the acute myocardial infarction.

tract (**LVtoCBcirc**). Furthermore, some physiological mechanisms through which the organism restores the corrupted blood flow were contemplated. The model already accounts for a decrease in urinary output, where the arterial blood pressure is dropped to restore the normal pressure within the systemic arteries (**LvtCBCirc**). In addition, the neurovegetative con-

trol (**SS**) over both heart beat frequency and the systemic arterial resistance (**VR**) have been represented. The node **SS** is sensitive to arterial blood pressure (**BP** is regarded as a manifestation of **LVtoCBcirc**) and it has, in turn, an impact on both arterial vascular resistance (**VR**) and heart frequency (**HF**). Arterial vascular resistance (**VR**) might increase the systemic arterial pressure (**LVtoCBcirc**). A node representing the persistence of sympathetic neurovegetative activation (SS_{pers}) has been added to account for the impairment of such a control mechanism when it lasts for too long. As such, SS_{pers} is influenced by the **SS** node, while it influences the **VR** node. Some variables show the status of the cardiovascular system to a medical observer, specifically, whether some of its tract is stagnant. Pulmonary congestion (**PulmCong**) might be the result of stagnation in the pulmonary venous tract (**RvtoPCcirc**), whereby peripheral congestion (**PeriphCong**) might be the result of stagnation in the systemic venous tract (**CBRVcirc**). These two phenomena manifest themselves respectively through shortness of breath (**Disp**) and pleural effusion (**PleuEff**) on one side, and pedal edema (**LPE**) on the other side. The first situation can be directly revealed by a low partial pressure of arterial oxygen (**O2pa**), whereas a severe reduction of blood perfusion gives rise to a fatal complicating condition known as shock (**Shock**). Heart failure is said to be cardiogenic, when it is the cardiac muscle (**Pump**) which is primarily impaired. Acute myocardial infarction (**AMI**) might be the cause of such damage, although in most instances it is not. As any other infarction, the condition is due to lack of arterial perfusion of the organ tissues. In the case of **AMI**, obstruction of coronary arteries (**CorObstr**) occurs, whose blood supply comes from the main arterial system (**LVtoCBcirc**). One important manifestation of coronary obstruction (**CorObstr**) is intense chest pain, called angina pectoris (**Angor**). Only when obstruction is severe and lasting is there infarction, it manifests through the increase of cardiac enzymes (**CardEnz**) in the blood stream and, in functional terms, the impairment of the cardiac pump (**Pump**). In turn, intense pain stimulates the neurovegetative system with an increment of sympathetic activity and, therefore, of blood pressure (**BP**) and heart frequency (**HF**). The number of states, together with their corresponding meaning, for each node of the CTBN model for the acute myocardial infarction (Figure 6.1) is reported

in Table 6.3.

Node	NS	MU	1	2	3
AMI	2		absent	present	
Angor	2		absent	present	
BP	3	mmHg	low [<80]	mid [80-106]	high [>106]
BV	3	mL	low [<4.41]	mid [4.41-5.39]	high [>5.39]
CardEnz	2		absent	present	
CBRVcirc	3	mmHg	low [<2]	mid [2-8]	high [>8]
CorObstr	3		normal	low	very low
Dispnr	2		absent	present	
HF	3	beat/min	low [<60]	mid [60-90]	high [>90]
LCI	3	mL/min	low [<4.2]	mid [4.2-6.3]	high [>6.3]
LCO	3	mL/min	low [<4.2]	mid [4.2-6.3]	high [>6.3]
LPE	2		absent	present	
LVtoCBCirc	3	mmHg	low [<80]	mid [80-106]	high [>106]
PCtoLVCirc	3	mmHg	low [<11]	mid [11-23]	high [>23]
PeriphCong	2		absent	present	
PleuEff	2		absent	present	
PulmCong	2		absent	present	
Pump	2		normal	reduced	
RCI	3	mL/min	low [<4.2]	mid [4.2-6.3]	high [>6.3]
RCO	3	mL/min	low [<4.2]	mid [4.2-6.3]	high [>6.3]
RVtoPCcirc	3	mmHg	low [<12]	mid [12-16]	high [>16]
O2pa	3	mmHg	<i>very low</i> [<60]	low [60-80]	normal [>80]
Shock	2		absent	present	
SS	3		parasympathic	normal	sympathic
SS-pers	2		inactive	active	
UO	3	um	low [<500]	mid [500-2,000]	high [$>2,000$]
VR	3	$\frac{dyn \cdot s}{cm^5}$	low [<900]	mid [900-1,200]	high [$>1,200$]
WI	3	mL	low [<500]	mid [500-2,000]	high [$>2,000$]

Table 6.3: Nodes and states for the acute myocardial infarction CTBN.

6.2.4 Inference

The CTBN model of acute cardiogenic heart failure, shown in Figure 6.1, has been exploited for the quantitative evaluation of the impact of observations whose medical meaning is well known and related to heart failure and acute myocardial infarction. According to the patient's evidence, the two disorders may coexist, or they can occur separately. Three scenar-

ios are described and analyzed to evaluate the model ability to determine medical conditions according to patient evidence. Likewise any evidence derived from patient monitoring will be referred in a time interval. In the following scenarios, only clinical manifestations entered the model, i.e. **BP**, **Disp**, **HF**, **Angor** and **LPE**. Instead, associated laboratory or imaging observations, like **CardEnz**, **O2pa**, **PulmCong** and **PleuEff**, can be predicted along with future relevant outcomes, like the occurrence of shock (**Shock**).

Scenario 1

The patient shows low blood pressure (**BP=low**), an increased heart frequency (**HF=high**), no chest pain (**Angor=absent**), shortness of breath (**Disp=present**) and pedal edema (**LPE=present**). All these manifestations last for 5 hours. Therefore, the CTBN model is inputted with the interval evidence;

[**BP=low**,**HF=high**,**Angor=absent**,**LPE=present**,**Disp=present**,**WI=mid**],

for the time interval from 0 to 5 hours, and the interval evidence [**WI=mid**], for the time interval from 5 to 6 hours. The occurrence of pedal edema (**LPE=present**) and shortness of breath (**Disp=present**) would make the doctor keen on the diagnosis of congestive heart failure, involving both the right and the left heart side. The absence of angor (**Angor=absent**) would make a diagnosis of acute myocardial infarction very unlikely. The doctor is aware that such a condition, if left untreated, could lead to shock. Conditionally on the above interval evidence, the posterior probability of shock (**Shock=present**) attains the highest peak at the end of the observations, reaching a posterior probability value equal to 0.59. A node belonging to the body internal state of the CTBN model for the acute myocardial infarction (Figure 6.1), i.e. the node **Pump** shows a reduced pump strength. Indeed, the posterior probability value associated with **Pump=reduced** is equal to 0.97 after one hour from the initial observations. Instead, the posterior probability value associated with the acute myocardial infarction (**AMI=present**) remains low (<0.01) during the whole period of interest, see Figure 6.2. This means that the patient is affected by primary congestive heart failure, whereas the adjective *primary* refers to a disease that is not the

secondary result of another disease.

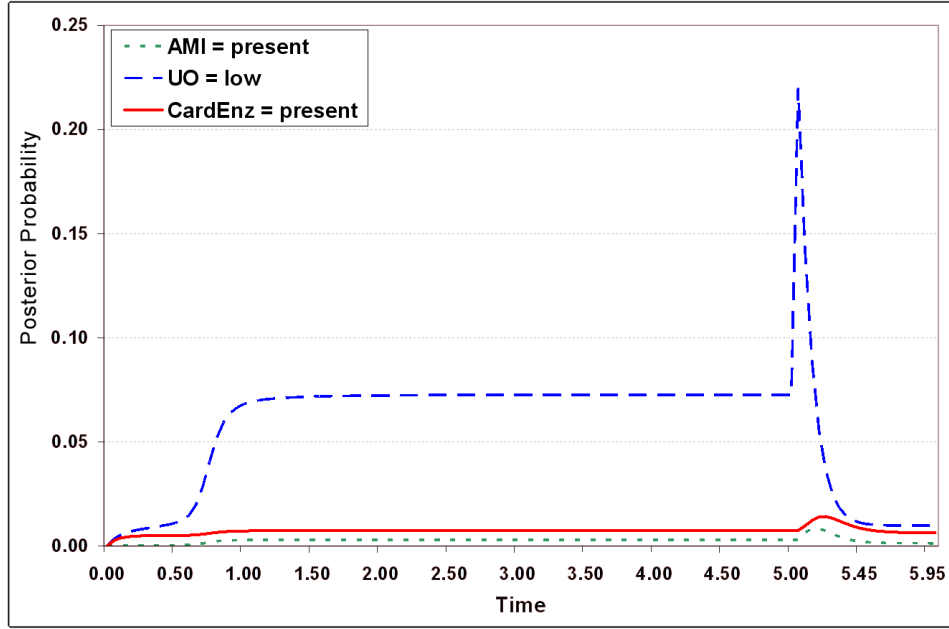


Figure 6.2: Posterior for AMI, UO and CardEnz under Scenario 1.

The increased probability value of the low urinary output (**UO=low**) (0.22 at the end of the observations) and the likely absence of cardiac enzymes (**CardEnz=present**) (<0.015) reinforces the above diagnosis see Figure 6.3.

Scenario 2

The patient shows normal blood pressure (**BP=mid**), an increased heart frequency (**HF=high**) and substernal chest pain (**Angor=present**). However, the patient does not show pedal edema (**LPE=absent**), nor shortness of breath (**Disp=absent**). These manifestations are supposed to last for 45 minutes. Therefore, the CTBN model is inputted with the following interval evidence:

[**BP=mid, HF=high, Angor=present, LPE=absent, Disp=absent, WI=mid**],
for the time interval from 0 to 45 minutes, and the interval evidence [**WI=mid**],
for the time interval from 45 minutes to 6 hours. Angor persisting (**An-**

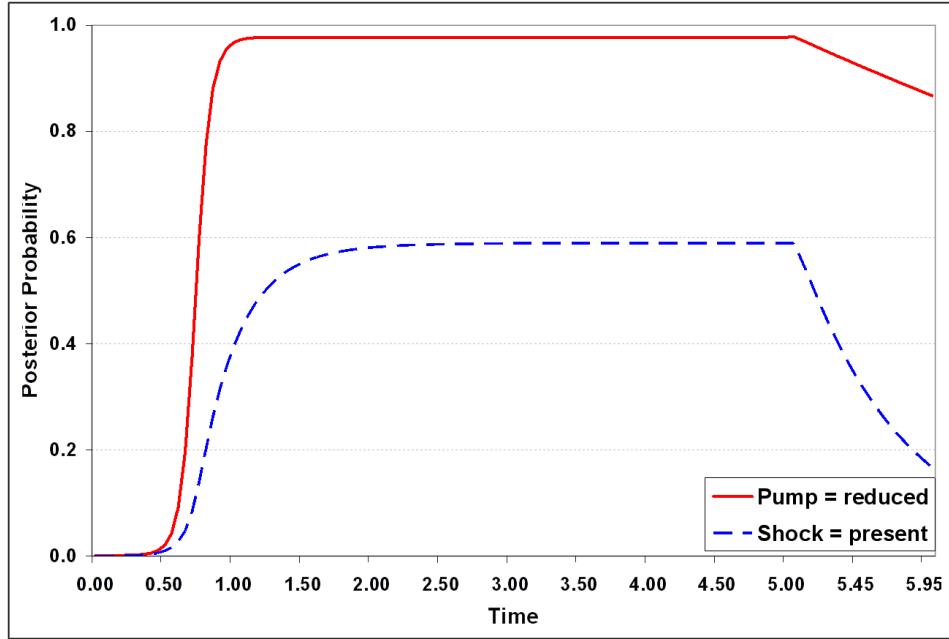


Figure 6.3: Posterior for **Pump** and **Shock** under Scenario 1.

gor=*present*) for more than half an hour, is a classical marker of acute myocardial infarction. Therefore, the pump strength is expected to be unaffected, since there are no signs of heart failure (**LPE**=*absent*, **Disp**=*absent*). The model predicts, even after one hour, a low probability of having shock (**Shock**=*present*) (0.0052). The probability of acute myocardial infarction (**AMI**=*present*) becomes as high as 0.45 after half an hour of persisting **angor**, but it decreases after the end of chest pain (from 0.49 to less than 0.02 after 15 minutes). Cardiac enzymes follow a similar evolution. The probability of a reduced pump strength (**Pump**=*reduced*) remains low (<0.10) during the time interval, likewise the probability of its associated manifestations, e.g. **UO** see Figure 6.4.

Scenario 3

The patient shows normal blood pressure (**BP**=*mid*), with increased heart frequency (**HF**=*high*) and chest pain (**Angor**=*present*). Like scenario 2, the patient does not show shortness of breath (**Disp**=*absent*) nor pedal edema

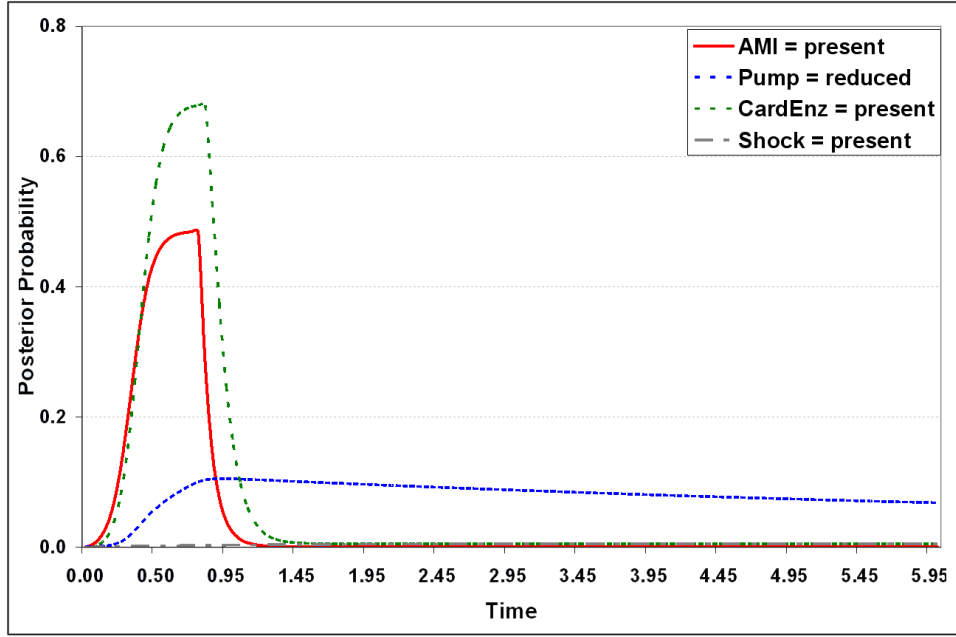


Figure 6.4: Posterior for AMI, Pump, CardEnz and Shock under Scenario 2.

(*LPE=absent*). However, the manifestations last only for 15 minutes. After this interval, the angina disappears (*Angor=absent*) for the next 15 minutes. Therefore, the CTBN model is inputted with the interval evidence:

[*BP=mid, HF=high, Angor=present, LPE=absent, Disp=absent, WI=mid*],
for the time interval from 0 to 15 minutes, the interval evidence

[*BP=mid, HF=high, Angor=absent, LPE=absent, Disp=absent, WI=mid*],
for the time interval from 15 to 30 minutes, and the interval evidence relative to node **WI** [*WI=mid*], for the time interval from 30 minutes to 6 hours. Physicians use to regard the occurrence of angor as a threatening condition because of its association with coronary obstruction, the cause of myocardial infarction. However, from a clinical point of view, when the chest pain does not persist for at least half an hour, the occurrence of acute myocardial infarction is unlikely. According to the model, heart failure and shock are predicted as unlikely events. In the following hour, the posterior probability of shock (*Shock=present*) remains low (<0.005), likewise the probability

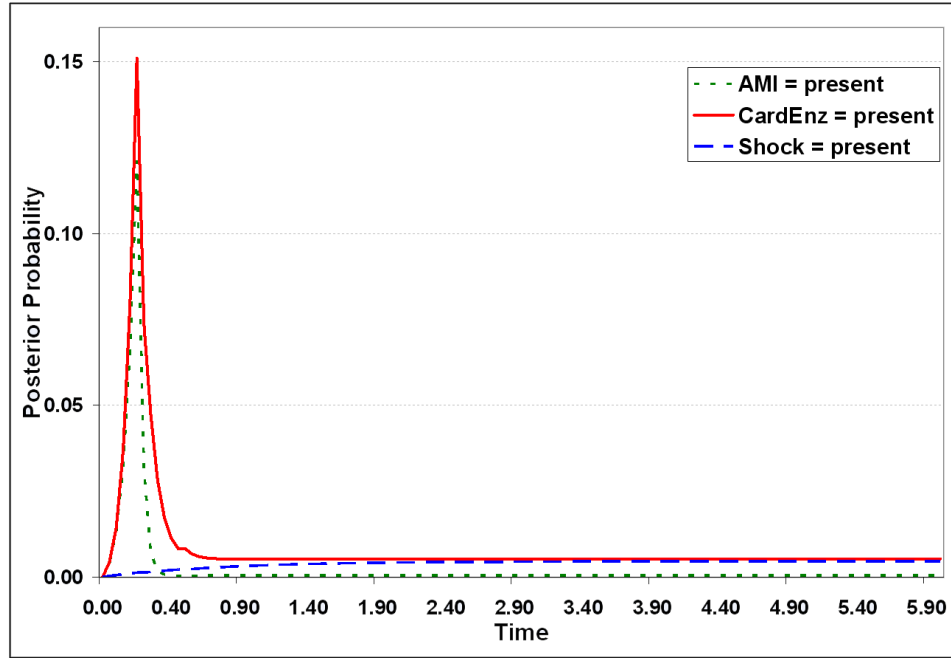


Figure 6.5: Posterior for AMI, CardEnz and Shock under Scenario 3.

of any other abnormal state, see Figure 6.5.

6.3 Discussion

The temporal dimension is an essential feature of medical reasoning and decision making. The diagnosis may take advantage from knowing the persistence of observations, and therapy may be optimized in the light of the likely future evolution of the medical disorder, anticipating complicating diseases. However, probabilistic inference in temporal models, especially Bayesian networks, is known to be excessively demanding, preventing both the assemblage of different sources of evidence in graphical terms and the exploitation of quantitative predictions while they are regarded as the most useful elements for the medical treatment of the patient. The epidemiological relevance of heart failure and the usefulness of accurate predictions in the correct management of such an evolving disorder is confirmed by other contributions addressed to the formal repre-

sensation of the disorder. Although methodologically different, they are all attempts to provide the problem with a quantitative analysis to be exploited in the medical practice. The *Seattle Heart Failure Model* is based on a survival model [Levy et al., 2006]. It is probably the first computer-based application to translate medications and devices that a heart failure patient receives into predicted years of survival. Similar considerations apply to other multivariate statistical models aimed at predicting mortality in patients with chronic heart failure [Brophy et al., 2004]. Bayesian reasoning and inference procedures have only recently gained popularity in the fusion of information obtained from different sources. Perhaps, their greatest potential in the clinical setting is to provide a pathophysiological interpretation of the monitored events. An influence diagram has been proposed to predict heart contractile power dysfunctions reflected in the condition of systolic heart failure [Fernandez et al., 2005]. Although the model is already structured as a decision support system, it is based on a static BN representation, this way skipping the complexity of inference along the temporal dimension. In spite of the prevalence of proportional hazards models as prognostic models in medicine, DBNs have been also proposed to take advantage of the causal and temporal nature of medical domain knowledge as elicited from domain experts [van Gerven et al., 2008]. To our knowledge, there has been only one attempt to model the evolution of heart failure by means of DBN [Hulst, 2006]. However, it is based on a time granularity of minutes, rather than seconds like in our application. This limit is likely to arise from the fact that BNs do not provide direct mechanisms for representing temporal dependencies, so any DBN representation, resulting from the assemblage of several BNs for each time of interest, tends to become rapidly intractable when applied to realistic domains [Kjaerulff, 1995]. The continuous time Bayesian network framework overcomes most of the difficulties presented above, making it possible to elaborate inference on complex medical problems. As such, it represents a significant improvement over DBNs. The above scenarios show to what extent the inference performed by a CTBN on a specific medical domain is achievable by means of ordinary hardware resources, while its consistency with physicians' judgments in the same domain is also shown. The comparison of Scenario 2

and Scenario 3 allows us to appreciate the impact of evidence known to be relevant for the occurrence of heart failure, although the reason of failure could be different. The first case study shows the typical consequences of a congestive heart failure, whereby the second patient shows symptoms of one potential cause of heart failure, i.e. acute myocardial infarction. In Scenario 1, the model correctly detects a primitive pump deficit as the cause of heart failure, anticipating shock as a likely future complication. Instead, there are no reasons to hypothesize a pump deficit as a secondary consequence of acute myocardial infarction. In Scenario 2, because the probability of heart failure is low and there are no symptoms of heart failure, the model correctly shows an uncomplicated acute myocardial infarction as the most likely diagnosis. Scenario 2 and Scenario 3 show the same set of the manifestations, but their comparison allows us to appreciate the impact of duration of pathological events. Physicians are aware that sub sternal chest pain is a symptom of coronary obstruction, whose impact on the myocardial tissue depends on the persistence of obstruction. Since an interval of 30 minutes is generally regarded as the trade off over which the occurrence of infarction becomes more likely than a simple angina episode, the model correctly discriminates the underlying diagnostic explanations of the two cases.

6.4 Conclusions

In this chapter the first clinical application of the recent developments in the research area of graphical modeling in continuous time have been described. This approach allows a direct representation of time and offers a valid computational machinery for medical inference. The predictions emerging from the three scenarios confirmed the heuristic power of the proposed framework and allowed a quantitative evaluation of the expected time before each variable changes its state. The proposed model has then the potential to be used for diagnostic purposes, as well as to develop a strategic plan to reduce the risk associated with each patient treatment. Additional improvements are needed to turn the CTBN on cardiogenic heart

failure into a practical medical tool. Quantitative parameters might be further tuned to achieve posterior probabilities that better fit with expectations derived from pathophysiological knowledge. This could be achieved by learning the CIMs directly from clinical data. The relevance of the continuous time Bayesian network model could be further increased through the embedding of the CTBN model into a DSS which assists the clinician to choose and apply the correct therapy. However, any additional features would preliminarily call for the computability of posterior probabilities of models as complex as the one presented. Thus, we anticipate the usefulness of CTBNs in clinical domains where, like in the case of heart failure, there is growing interest in quantitative predictions.

Chapter 7

Decision Making under Uncertainty

We now move from the task of simply reasoning under uncertainty, to the task of deciding how to act in the world. In a decision-making setting, an agent has a set of possible actions and has to choose between them. Each action can lead to one of several outcomes, which the agent can prefer to different degrees. More simply, the outcome of each action is known with certainty. In this case, the agent must simply select the action that leads to the outcome that is most preferred. Even this problem is far from trivial, since the set of outcomes can be large and complex and the agent must weigh different factors in determining which of the possible outcomes is most preferred. Even more difficult is the decision-making task in situations where the outcome of an action is not fully determined. In this case, we must take into account both the probabilities of various outcomes and the preferences of the agent between these outcomes. Here, it is not enough to determine a preference ordering between the different outcomes. We must be able to ascribe preferences to complex scenarios involving probability distributions over possible outcomes [Koller and Friedman, 2009].

The framework of *decision theory* provides a formal foundation for this type of reasoning. This framework requires that we assign numerical *utilities* to the various possible outcome, encoding the agent's preferences. The principle of *maximum expected utility*, which is the foundation for decision

making under uncertainty, is defined. After the description of the basic concepts of the decision theory, we propose structured representation for decision-making problems and algorithms that exploit this structure when addressing the computational task of finding the decision that maximizes the expected utility. We begin by describing *decision trees*, a simply and intuitive representation that describes a decision making situation in terms of the scenario that the decision maker might encounter. This representation, unfortunately, scales up only to fairly small decision tasks; still, it provides a useful basis for much of the later developments. We describe *Influence Diagrams*, which extend Bayesian Networks by introducing decisions and utilities. We then discuss an approach for approximate representation of *optimal strategies* in influence diagrams.

7.1 Decision Theory

A Bayesian network provides a model of the world that can be used in making decisions. The typical situation is that we observe some of the variables in the domain and based on these observations we make an inquiry to the Bayesian network about some other set of variables (probability updating). The result of the query is in turn used in the subsequent decision-making process. This type of application of Bayesian networks can be taken one step further, so that rather than keeping the model separated from the decision making process, you could combine these two parts. That is, not only does the final model reveal the structure of the decision problem, but it can also be used to give advice about the decisions. In the simple situation in which only a single decision is to be made, the Bayesian network can readily be extended to reflect the structure of the decision problem; it is possible to use the model to give advice on the decision D .

The general situation with one decision variable is as described in Figure 7.1. There is a Bayesian network structure with chance nodes and directed links. The network is extended with a single decision node D that may have an impact on the variables in the structure. In other words, there may be a link from D to some chance nodes. Furthermore, there is a set of

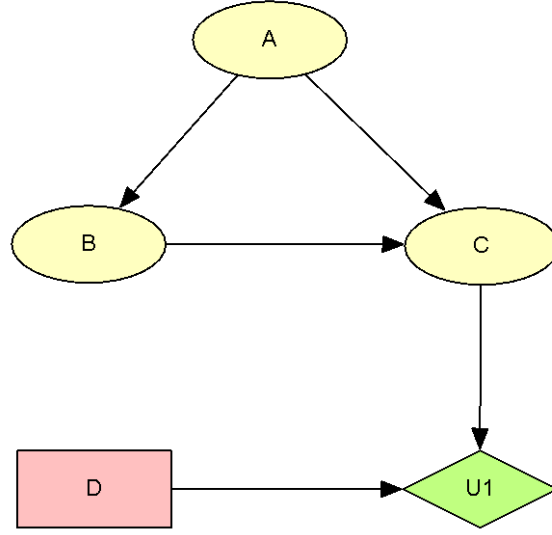


Figure 7.1: Graphical representation of a decision problem with A, B, C chance nodes. The variable D is a decision variable. The variable U is the utility node.

utility functions, $U_1, \dots, U_n$, over domains $X_1, \dots, X_n$. The task is to determine the decision that yields the *maximum expected utility*. Thus, if none of the utility nodes contain D in the domain, then with evidence e we calculate:

$$EU(D|e) = \sum_{X_1} U_1(X_1)P(X_1|D, e) + \dots + \sum_{X_n} U_n(X_n)P(X_n|D, e) \quad (7.1)$$

and a state d maximizing $EU(D = d|e)$ is chosen as an optimal decision. When D is contained in the domain of a utility node, such as U_1 in Figure 7.1, then we should perform the summation only over $X_1 \setminus \{D\}$, and accordingly, we should use the probability distribution $P(X_1 \setminus \{D\}|D, e)$. A requirement of the method described above is that the decision problem contains only a single decision. The various decisions should be evaluated on the basis of the usefulness of their consequences. And the usefulness is assumed to be measured on a numerical scale called *utility scale*, and if several kinds of utilities are involved in the same decision problem, then the scale have a common unit [Jensen and Nielsen, 2007].

7.2 Structured Decision Problems: Decision Trees

A classical way of representing decision problems with several decisions is with decision trees. A decision tree is a model that encodes the structure of the decision problem by representing all possible sequences of decisions and observations explicitly in the model. The non-leaf nodes are decision nodes (rectangular boxes) or chance nodes (circles or ellipses), and the leaves are utility nodes (diamond shaped). The arcs in the tree have labels. An arc from a decision node is labeled with the action chosen, and an arc from a chance node is labeled by a state. A decision tree is read from the root downward. When you pass a decision node, the label suggests you what the decision is, and when you pass a chance node, the label tells you the state of the node. If a chance node precede a decision node, then the chance node is said to be observed before the decision is made. Hence the sequence in which we visit the nodes corresponds to the sequence of observations and decisions. We assume no-forgetting: when a decision is to be taken, the decision maker knows all the labels on the path from the root down to the current position in the decision tree. We adopt the shorthand path for the set of labels from the root to a position in the tree. A sequence in a decision scenarios is the sequence of observations and decisions that we read on the path from the root to the leaf. A decision trees must to be complete: from a chance node there must be a link for each possible state, and from a decision node there must be a link for each possible action. This also means that a decision tree specifies all the possible scenarios in the decision problem. The quantitative part of a decision tree consists of utilities and probabilities. Each leaf has a utility value attached to it. This utility reflects the utility of the decision scenario identified by the path from the root to the leaf in question. For the chance nodes, we associate a probability with each of the links starting from them. As in all decision problems, a solution to a decision tree is a strategy that suggests how you should act at the various decision nodes. Strategies are compared based on their expected utilities, and finding an optimal strategy amounts to finding a strategy with maximum expected utility; such a strategy is not necessarily unique. By assigning to each node in the decision tree a value

corresponding to the maximum expected utility achievable at that node, an optimal strategy will pick an action leading to a child of maximum value. Looking at the end of the decision tree, one sees that the value of a leaf node is simply the utility assigned to that node. If we go one step further up the tree, then the value of a decision node D is the maximum value associated with its children/leaves, since D is under full control. For a chance node, its value corresponds to the utility you can expect to achieve from that point in the decision tree: the value is the sum of the utilities of the leaves weighted with the probabilities of their *outcomes*. If X is a decision node, we assign it the maximum of the children's values, and if X is a chance node, we assign the weighted sum. In order to calculate an optimal strategy and the maximum expected utility, the procedure known as 'average-out and fold-back' can be applied, starting with nodes that have only leaves as children. If the node is a chance node A , the expected utility for A is calculated. Each child of A is an outcome o and has a utility $U(o)$ attached, and the link has a probability $P(A = a)$. The product $U(o)P(A = a)$ from each child can be calculated, and their sum is attached to A . If the node is a decision node D , each child of D has an (expected) utility attached. Choose a child with maximal expected utility and attach the value to D . This is done repeatedly until the root is reached. The resulting value for the root is the expected utility if you adhere to the strategy of always maximizing the expected utility, and the paths from root to leaves following the visited links represent an optimal strategy for the decision problem. The expected utility of an optimal strategy Δ is the sum of the utilities of the possible outcomes o (the leaves in the decision tree) weighted by the probability of the path down to o under the strategy Δ :

$$EU(\Delta) = \sum_o U(o)P(o|\Delta). \quad (7.2)$$

In general, this procedure can be used for calculating the expected utility of any strategy; the identification of an optimal strategy can also be formulated as:

$$\Delta = \operatorname{argmax}_{\Delta'} EU(\Delta'). \quad (7.3)$$

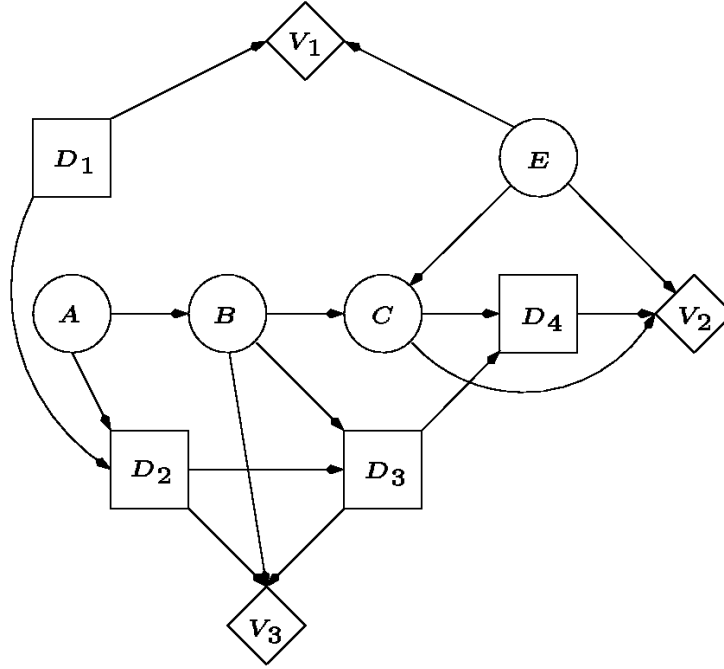


Figure 7.2: The figure depicts an ID with four chance nodes A , B , C and E , four decision nodes D_1 , D_2 , D_3 , D_4 and three value nodes V_1 , V_2 and V_3 .

7.3 Structured Decision Problems: Influence Diagrams

The *Influence Diagram* framework was developed as a computer aided modeling tool by Miller [Miller et al., 1976] and later by Howard and Matheson [Howard and Matheson, 1981]. The ID serves as an efficient modeling tool for symmetric decision problems with several decisions and a single decision maker. In this particular context we will take into account the ID with discrete variables. An Influence Diagram can be seen as a Bayesian Network (BN) augmented with decision nodes and value nodes, where value nodes have no descendants. Thus, an influence diagram is a directed acyclic graph $G = (\mathcal{U}, \mathcal{E})$, where the nodes $\mathcal{U}$ can be partitioned into three disjoint subsets; *chance nodes* $\mathcal{U}_C$, *decision nodes* $\mathcal{U}_D$ and *value nodes* $\mathcal{U}_V$.

An example of influence diagram [Nielsen, 2001] is shown in Figure 7.2. In this chapter we will use the concept of node and variable interchangeably

if this does not introduce any inconsistency. We will also assume that no *barren nodes* are specified by the influence diagram since they have no impact on the decisions [Shachter, 1986]; a chance node or a decision node is said to be barren if it has no children, or if all its descendants are barren. Furthermore, in an influence diagram we have a total ordering of the decision nodes indicating the order in which the decisions are made (the ordering of the decision nodes is traditionally represented by a directed path which includes all decision nodes) [Nielsen and Jensen, 2004]. With each chance variable and decision variable X we associate a finite *state space* $sp(X)$, which denotes the set of possible outcomes/decision options for X . For a set $\mathcal{U}'$ of chance variables and decision variables we define the state space as $sp(\mathcal{U}') = sp(X) | X \in \mathcal{U}'$, where $A \times B$ denotes the Cartesian product of A and B . The uncertainty associated with each chance variable C is represented by a conditional probability potential $P(C|pa(C)) : sp(\{C\} | pa(C)) \rightarrow [0; 1]$, where $pa(C)$ denotes the parents of C in the influence diagram. The domain of a conditional probability potential $\phi_C = P(C|pa(C))$ is denoted $dom(\phi_C) = \{C\} \cup pa(C)$. The decision maker's preferences is described by a multi-attribute utility potential; the set of value nodes $\mathcal{U}_V$ defines the set of utility potentials which appear as additive components in the multi-attribute utility potential. Each utility potential indicates the local utility for a given configuration of the variables in its domain. The domain of a utility potential ψ_V is denoted $dom(\psi_V) = pa(V)$, where V is the value node associated with ψ_V . Analogously to the concepts of variable and node we shall sometimes use the terms value node and utility potential interchangeably. A realization of an influence diagram ID is an attachment of potentials to the appropriate variables in ID , i.e., a realization is a set $\{P(C|pa(C)) | C \in \mathcal{U}_C\} \cup \{\phi_V(pa(V)) | V \in \mathcal{U}_V\}$. So, a realization specifies the quantitative part of the model whereas the influence diagram constitutes the qualitative part. The arcs in an influence diagram can be partitioned into three disjoint subsets, corresponding to the type of node they go into. Arcs into value nodes represent functional dependencies by indicating the domain of the associated utility potential. Arcs into chance nodes, termed dependency arcs, represent probabilistic dependencies, whereas arcs into decision nodes, termed informational arcs, imply information precedence;

if there is an arc from a node X to a decision node D , then the state of X is known when decision D is made. Let $\mathcal{U}_C$ be the set of chance variables and let $\mathcal{U}_D = \{D_1, D_2, \dots, D_n\}$ be the set of decision variables. Assuming that the decision variables are ordered by index, the set of *informational arcs* induces a partitioning of $\mathcal{U}_C$ into a collection of disjoint subsets $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_n$. The set $\mathcal{C}_j$ denotes the chance variables observed between decision D_j and D_{j+1} . Thus the variables in $\mathcal{C}_j$ occur as immediate predecessors of D_{j+1} . This induces a partial order $\prec$ on $\mathcal{U}_C \cup \mathcal{U}_D$, i.e., $\mathcal{C}_0 \prec D_1 \prec \mathcal{C}_1 \prec \dots \prec D_n \prec \mathcal{C}_n$. The set of variables known to the decision maker when deciding on D_j is called the *informational predecessors* of D_j . By assuming that the decision maker remembers all previous observations and decisions, we have informational predecessors of $D_i \subseteq$ predecessors of D_j (for $D_i \prec D_j$) and in particular, predecessors of D_j are the variables that occur before D_j under $\prec$. This property is known as *no-forgetting* and from this we can assume that an influence diagram does not contain any no-forgetting arcs, i.e., $pa(D_i) \cap pa(D_j) = \emptyset$ if $D_i \neq D_j$.

When evaluating an influence diagram we identify a strategy for the decisions involved; a strategy can be seen as a prescription of responses to earlier observations and decisions. The evaluation is usually performed according to the *maximum expected utility* principle, which states that we should always choose an alternative that maximizes the expected utility.

Definition 4. [Nielsen and Jensen, 2004] Let ID be an influence diagram and let $\mathcal{U}_D$ denote the decision variables in ID . A strategy is a set of functions $\Delta = \{\delta_D \mid D \in \mathcal{U}_D\}$, where δ_D is a policy given by:

$$\delta_D : sp(pred(D)) \rightarrow sp(D).$$

A strategy that maximizes the expected utility is termed an *optimal strategy*, and each policy in an optimal strategy is termed an *optimal policy*.

In general, the optimal policy for a decision variable D_k is given by:

$$\begin{aligned} & \delta_{D_k}(\mathcal{C}_0, D_1, \dots, D_{k-1}, \mathcal{C}_{k-1}) = \\ & \underset{\mathcal{C}_k}{\operatorname{argmax}}_{D_k} \sum P(\mathcal{C}_k, \mathcal{C}_0, D_1, \dots, \mathcal{C}_{k-1}, D_k) \rho_{D_{k+1}} \end{aligned} \quad (7.4)$$

where $\rho_{D_{k+1}} = \sum_{U \in \mathcal{U}_V} \psi_V$ if $k = n$ otherwise $\rho_{D_{k+1}}$ is the maximum expected utility potential for decision D_{k+1} :

$$\rho_{D_{k+1}}(\mathcal{C}_0, D_1, \dots, D_k, \mathcal{C}_k) = \max_{D_{k+1}} \sum_{\mathcal{C}_{k+1}} P(\mathcal{C}_{k+1} | \mathcal{C}_0, D_1, \dots, \mathcal{C}_k, D_{k+1}) \rho_{D_{k+2}} \quad (7.5)$$

As the domain of a policy function grows exponentially with the number of variables in the past, it is important to weed out variables irrelevant for the decision [Nielsen and Jensen, 2004].

7.4 Information enhancement

The main source of complexity problems for large influence diagrams is that the last decisions have intractably large spaces of past information. Usually, it is not a problem when you reach the last decisions; but when calculating optimal policies for the first decisions, you have to consider all possible future information scenarios. This is *the curse of knowing that you shall not forget*. The usual approach for addressing this problem is to reduce the information through assuming that you do forget something (LIMID, [Lauritzen and Nilsson, 2001]), or to abstract the information through introducing new nodes [Jensen, 2008]. In this part of the chapter we take the opposite approach, namely to assume that you know more in the future than actually will be the case. We call the approach *information enhancement*. We reduce the future information scenarios by adding information links. We present a systematic way of determining information links to add [Jensen and Gatti, 2010].

7.4.1 Introduction

As opposed to decision trees, influence diagrams are easy to enter to a computer. Hence the hard job is to establish a solution, a set of policies, one for each decision. There are several algorithms for solving IDs [Shachter, 1986, Shenoy, 1992, Jensen et al., 1994], but the principle behind them all is dynamic programming starting with the last decision. That is, first an

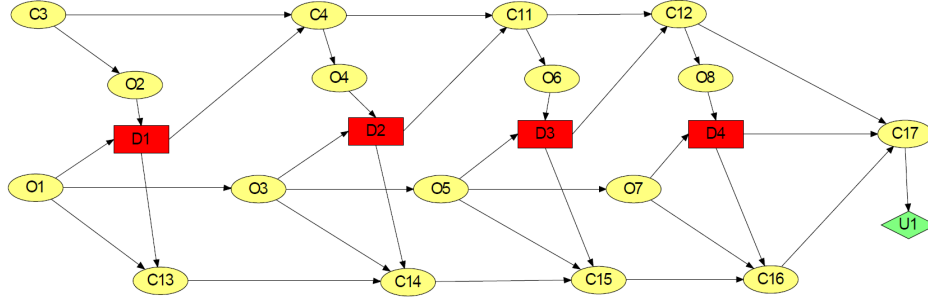


Figure 7.3: An influence diagram over variables with ten states.

optimal policy for the last decision is determined. Next, this policy is represented somehow, and the optimal policy for the second last decision is determined by using the policy for the last decision for forecasting the future. The way it is done is *variable elimination*: all variables are successively removed from the graph, and when a variable A is removed, the resulting graph will hold a link between any pair of A 's neighbors.

The solution phase may be very demanding with respect to time and space, but it is an off-line activity where you are not bound by tough constraints. The complexity problem you meet in the solution phase is that when you eliminate a variable A you have to work with a joint table over A and its neighbors, and that set of nodes may become (almost) too large to handle.

The next task is to *represent* the solution. The policies in the solution may have very large domains. Take for example the last decision in a sequence of ten. Then the policy δ_{10} is a function whose domain may include all previous observations and decisions.

For illustration, look at Figure 7.3. The domain for δ_4 contains 11 variables with ten states ($\{O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8, D_1, D_2, D_3\}$). This means that variable elimination will have to deal with tables with 10^{11} entries. It is an off-line activity, and you may succeed by spending much time, space, and exploiting sophisticated machines and/or cloud computing. Although it may seem intractable to represent δ_4 for fast on-line access, it is not a problem: the ID itself is a very compact representation of a policy for the

last decision. When you are about to take the decision D_4 , you know the state of the information variables, and it is an easy computational task to find the optimal decision.

The problem concerns with the first decision. When taking the first decision you must anticipate what you will do when taking the last decision. However, you do not know the information available at that time, and therefore you in principle have to work with the joint probability of all the unknown information variables (including future decisions). This is what we call *the curse of knowing that you do not forget*.

Usually the domain of the first decision is not extremely large, so you may off-line compute an optimal policy, which can be stored for fast access. We shall address the decisions in between, and we construct influence diagram representations, where policies of future decisions are approximated through reduction of their domain (see Figure 7.4 for an illustration).

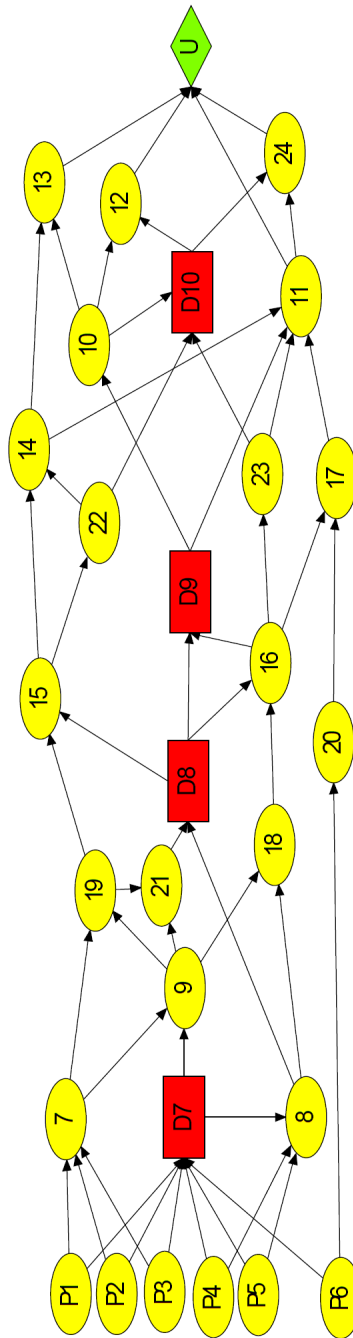


Figure 7.4: The general situation. You are in the middle of a series of decisions (D_7); you have collected much information, and now you have to anticipate a future decision (D_{10}).

If the ID in Figure 7.4 is used to represent the policy δ_7 , then the nodes P_1 to P_6 are known. That is, the state of these nodes are entered before the solution algorithm is started, and they do not contribute to the space complexity of the solution algorithm. The problem for the situation in Figure 7.4 is twofold: the space of the past for D_7 is too large such that δ_7 cannot be represented as a look-up table, and the space of future information relevant for D_{10} is so large that an on-line solution of the ID is not tractable. The problem has previously been addressed by an approach, which can be characterized as *information abstraction*: you aim at determining a set of variables which serve as an abstraction of the actual information. This may be done with the LIMID approach [Lauritzen and Nilsson, 2001], where it is assumed that some information will be forgot in the future, or it may be done through introduction of new nodes (like history nodes) through which the information is passed through [Jensen, 2008].

In this chapter we take the opposite approach, which we call *information enhancement*: we assume the decision maker to be more informed than actually will be the case.

7.4.2 Information enhancement approach

Our information enhancement approach consists of determining a small set of variables, which if known would overwrite the actual information. We shall use the terms disclosed and closed for variables with known state and unknown state, respectively. The idea behind information enhancement is to find a cut set S which d-separates the rest of the information from the relevant utilities. When S has been determined, we assume it to be disclosed when taking the future decision. We shall say that the new information nodes are enhanced. To illustrate the approach, consider a finite horizon Partially Observable Markov Decision Process (POMDP) [Drake, 1962] (see Figure 7.5).

As the nodes C_1 to C_7 may be a compound of several variables, and the observed nodes may also be a set of variables, we may assume that all the chance variables have 50 states. Now, consider the decision D_3 . The past is too large for a direct representation of δ_3 , and the influence diagram

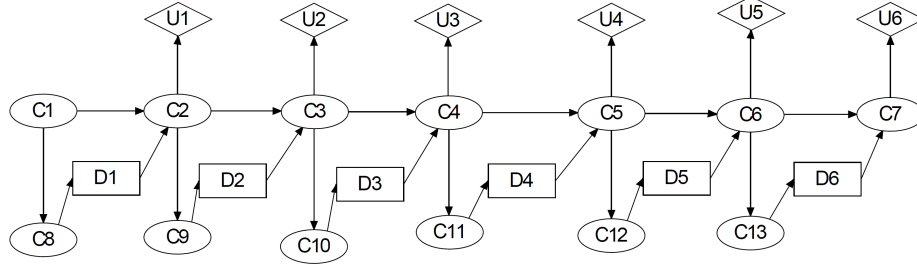


Figure 7.5: A POMDP

with the past of D_3 instantiated is also too complex. We can approximate δ_3 by approximating δ_6 through enhancing C_6 (see Figure 7.6), and the largest policy domain when solving the ID will contain four variables (δ_5 has the domain $\{D_3; C_{11}; D_4; C_{12}\}$).

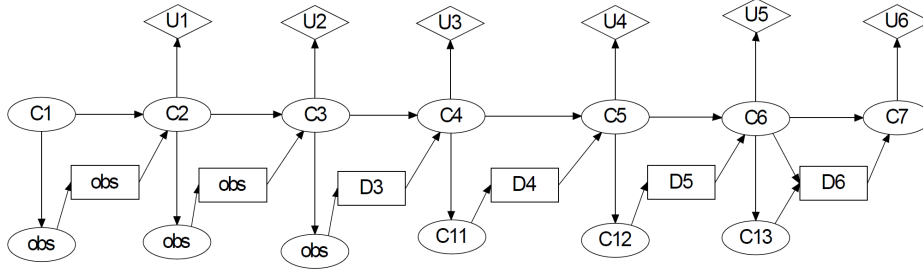


Figure 7.6: C_6 is enhanced for D_6 . With the past of D_3 instantiated, the largest policy domain contains four variables.

You may also choose to approximate δ_5 through enhancing C_5 (Figure 7.7), and the largest clique when solving the ID contains three variables ($\delta_6(C_5, D_5, C_{13})$).

As a small test of the approach we tried the three structures above (with only binary variables) with three different arbitrary parameter settings, and with utilities after each move as well as with utility after the last move, only. We looked at the policy δ_3 , and for all cases, the optimal policy was the same as the approximated one.

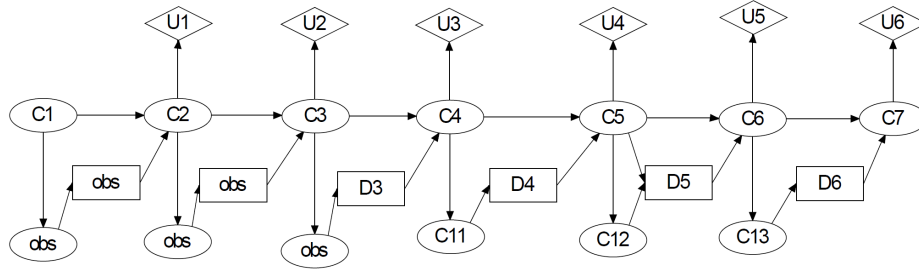


Figure 7.7: C_5 is enhanced for D_5 . Now the largest domain contains three variables.

7.4.3 Maximize uncertainty

Consider the general situation as described in Figure 7.4. If we wish to approximate D_{10} by information enhancement, we can enhance the pair (C_{11}, C_{13}) as well as (C_{11}, C_{14}) - blocking for everything but C_{10} . (See Figure 7.8). When discussing IDs we shall use the terms 'variable' and 'node' interchangeably. The node 14 is further away from the utility node than 13, and therefore, disclosing 13 will give you more certainty of the expected utility than would disclosing 14. This means that adding the information 14's state brings you closer to the actual knowledge at the time of deciding D_{10} than would adding the information of 13's state, and enhancing $(11; 14)$ is a better approximation than enhancing $(11; 13)$. We have performed a small experiment with the ID in Figure 7.4 and approximated δ_7 with the optimal policy from Figure 7.8. All nodes were binary. Out of the 64 configurations of the domain, the policies coincided on 61 cases. For one case the approximated policy has a tie between the correct decision and another one, in the two other cases, the difference in EU between the correct and the approximated decision was 0.001 on a value around 50 (on a scale from 0 to 100).

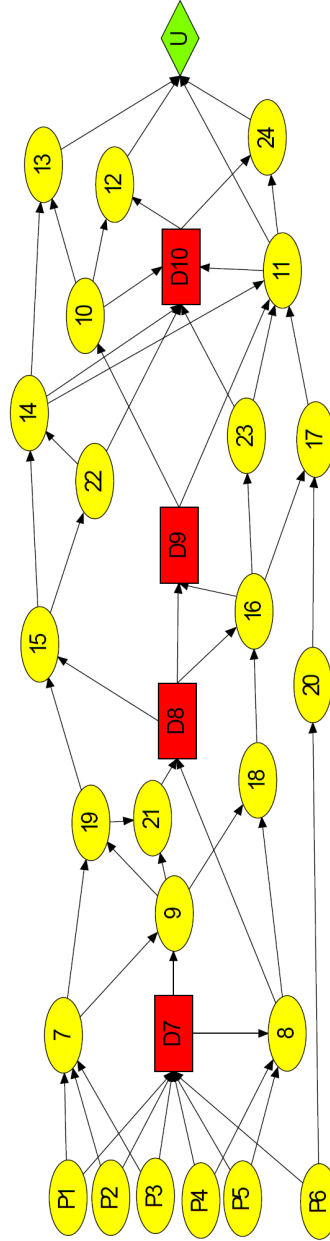


Figure 7.8: The ID in Figure 7.4 with the nodes 11 and 14 enhanced.

7.4.4 Border and Frontier

In general, we have two decision nodes D_i and D_j ($i < j$) in an influence diagram. The set of disclosed variables at the time of deciding D_i is denoted $\mathcal{P}$. We should index the set with i , but for notational convenience we will skip the indices i and j . The set of nodes becoming disclosed between deciding D_i and D_j (including D_i) is denoted $\mathcal{Inf}$. $\mathcal{P}$ and $\mathcal{Inf}$ are the disclosed nodes. With $i = 7$ and $j = 10$ in Figure 7.4 we have that $\mathcal{P}$ is the nodes P_1 to P_6 and $\mathcal{Inf} = \{D_7; D_8; D_9; 8; 10; 16; 21; 22; 23\}$. The set of descendants of D_j is denoted $\mathcal{D}$. Only the utility nodes in $\mathcal{D}$ are relevant for D_j . They are denoted $\mathcal{U}$. In Figure 7.4, $\mathcal{D} = \{12; 24; U\}$ and $\mathcal{U} = \{U\}$. The scene is now that the utility nodes of interest are $\mathcal{U}$, and we look for closed nodes, which if disclosed would turn some nodes in $\mathcal{I} \setminus \mathcal{I}$ irrelevant. That is, we search for cut sets $\mathcal{C}$ such that $\mathcal{U}$ is d-separated from $\mathcal{I} \setminus \mathcal{I}$ given $\mathcal{C}$ (we define d-separation such that nodes from $\mathcal{Inf}$ are allowed in $\mathcal{S}$). The chance nodes in $\mathcal{D}$ can not be used in such cut sets as this would create a directed cycle. The basic idea is to establish two cut sets, the *border* and the *frontier*. The border is the smallest cut set of non- $\mathcal{D}$ chance nodes closest to $\mathcal{U}$.

Definition 5. A node $X \notin \mathcal{D}$ belongs to the border if:

- X is a parent of an element of $\mathcal{D}$
- There is an active path from $\mathcal{Inf}$ to X

The set of border nodes is denoted by $\mathcal{B}$.

In Figure 7.4 the border consists of the nodes $\{10; 11; 13\}$. As none of the descendants of $\mathcal{B}$ are disclosed before deciding D_j we have:

Proposition 1. $\mathcal{Inf}$ is d-separated from $\mathcal{U}$ given $\mathcal{B}$.

Knowing that $\mathcal{B}$ is a cut set, you may go backwards from $\mathcal{B}$ in the network to create new cut sets. Actually, all chance nodes on active paths from $\mathcal{Inf}$ to $\mathcal{B}$ may be part of a cut set. The task is to identify the relevant part of the network and for this relevant part to identify good cut sets for information enhancement.

Definition 6. A network is regular if there is no active path from $\mathcal{Inf}$ to $\mathcal{U}$ involving a converging connection over a node in $\mathcal{P}$ or with a descendant in $\mathcal{P}$.

The network in Figure 7.4 is regular, and in the following sections we assume the network in consideration to be regular.

Definition 7. The set of information holders, $\mathcal{I}$, consists of all closed nodes with a directed path of closed nodes to $\mathcal{Inf}$.

For the network in Figure 7.4 we have $\mathcal{I} = \mathcal{Inf} \cup \{7; 8; 9; 15; 18; 19\}$.

Definition 8. A node in $\mathcal{I}$ that has a directed path of closed nodes to $\mathcal{U}$ and with no intermediate nodes in $\mathcal{I}$ is said to belong to the frontier of D_j . The set of frontier nodes is denoted by $\mathcal{F}$.

In Figure 7.4 the frontier of D_{10} consists of the nodes $\{10; 15; 16; 22; 23; D_9\}$.

Theorem 1. $\mathcal{I}$ is d -separated from $\mathcal{U}$ given $\mathcal{F}$.

Proof. Let $V_0 \in \mathcal{I}, U \in \mathcal{U}$, and let $\langle V_0, \dots, V_k \rangle$ be an active path given $\mathcal{F}$. Assume that $\langle V_0, \dots, V_k \rangle$ contains a converging connection, and let $V_{s-1} \rightarrow V_s \leftarrow V_{s+1}$ be the last converging connection on the path from V_0 to U . As $V_s \notin \mathcal{P}$ nor has a descendant in $\mathcal{P}$, V_s or one of its descendants is disclosed, and hence $V_s \in \mathcal{I}$. Therefore, also $V_{s+1} \in \mathcal{I}$. When you follow the path toward U you will meet a diverging connection $V_{t-1} \leftarrow V_t \rightarrow V_{t+1}$. Then $V_t \in \mathcal{F}$, and the path is not active. Note that you will meet a diverging connection at the latest when you reach $V_k \rightarrow U$. We conclude that there is no converging connections on the path. Assume that the first link is $V_0 \leftarrow V_1$. Then, follow the path until you reach a diverging connection. As there are no converging connections on the path, there must be exactly one diverging connection $V_{s-1} \leftarrow V_s \rightarrow X$. Then $V_s \in \mathcal{F}$ and the path is not active. To conclude: the active path is directed from V_0 to U , and it cannot contain intermediate nodes from $\mathcal{F}$. Therefore $V_0 \in \mathcal{F}$. $\square$

From the proof above we can conclude that information from $\mathcal{Inf}$ flows to U through a path against the direction of the links followed by a path along the links. The node, where the direction of the flow turns, is a frontier node.

7.4.5 Finding the border and the frontier

There are two obvious candidate sets for enhancement, namely $\mathcal{B}$ and $\mathcal{F}$. They are determined through a sequence of graph searches (for example breath-first search). First you determine $\mathcal{D}$ and $\mathcal{U}$, by starting a breath first search from the decision D_j . All chance nodes reached are labeled D . They are the elements of $\mathcal{D}$, and the utility nodes are the elements of $\mathcal{U}$. The nodes in $\mathcal{D}$ cannot be enhanced as this will introduce a directed cycle. The non- $\mathcal{D}$ parents of the nodes in $\mathcal{D} \cup \mathcal{U}$ are the candidate border nodes, and they are labeled CB . Next, start a backwards breath-first search from each of the decision nodes $D_{i+1}, \dots, D_j$.

That is, you follow the edges opposite to their direction. You stop when you meet a previous decision node or a node in $\mathcal{P}$. Each node you meet is labeled with an $\mathcal{I}$. Perform a backwards breath-first search from the nodes of $\mathcal{D}$. When you meet a node X with label I you give it the label F , and break the search behind X . Finally, perform a breath-first search from $\mathcal{F}$. When you meet a node X with label CB , change the label to B and stop searching behind X . The various labels for the ID in Figure 7.4 are given in Figure 7.9.

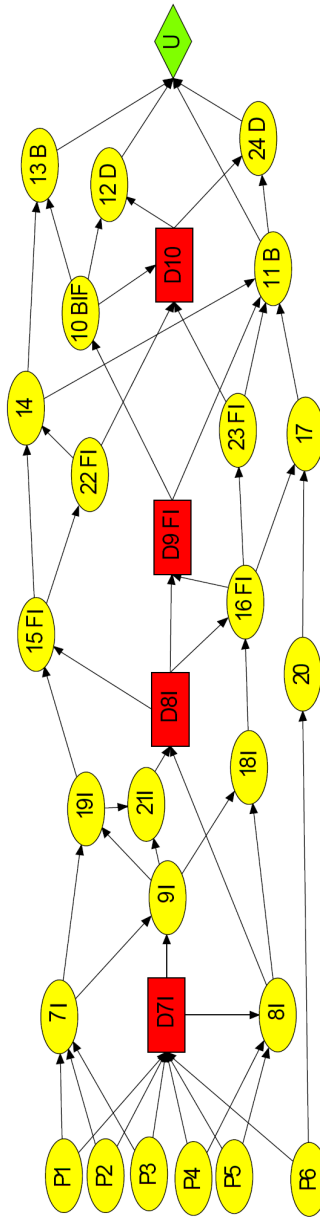


Figure 7.9: The ID marked after the search algorithm. "D" indicates that the node cannot be enhanced; "I" indicates nodes with information to transmit; "B" indicates border; "F" indicates frontier.

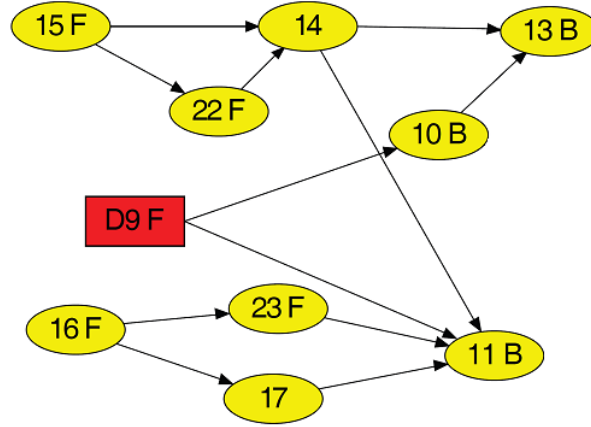


Figure 7.10: The free graph for the ID in Figure 7.4

7.4.6 Cut sets between frontier and border

There may be other candidate sets for enhancement than $\mathcal{B}$ and $\mathcal{F}$. Actually, any set of nodes which d-separates the frontier from the border can be used for enhancement.

Proposition 2. Let $\langle V_0, \dots, V_k \rangle$ be an active path with $V_0 \in \mathcal{F}$ and $V_k \in \mathcal{B}$. Then the intermediate nodes cannot be in $\mathcal{D}$ nor in $\mathcal{I}$, and the path is directed from V_0 to V_k .

Proof. The proof of Theorem 1. □

Definition 9. (Free graph). The subgraph $\mathcal{G}$ consisting of $\mathcal{F}$, $\mathcal{B}$ and all nodes on a directed path from a node in $\mathcal{F}$ to a node in $\mathcal{B}$ is called the free graph (see Figure 7.10).

The proposition yields that we can use any set in $\mathcal{G}$ which d-separates $\mathcal{F}$ from $\mathcal{B}$. Unfortunately, finding all possible cut sets may for a large $\mathcal{G}$ be intractable [Provan and Ball, 1983]. If that is the case, the following heuristics can in polynomial time provide a set of very good candidates for information enhancement. Note that you cannot just perform a flow analysis on the directed graph $\mathcal{G}$. In Figure 7.10, for example, the set $\{10; 11; 15; 22\}$ does not block for the information coming from 16 or 23.

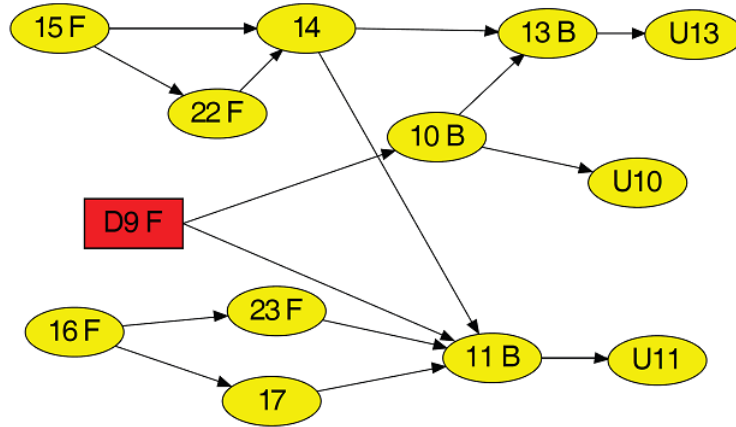


Figure 7.11: The extended free graph for the ID in Figure 7.4

7.4.7 Cut set heuristics

To indicate that the information is flowing to $\mathcal{B}$, you extend the free graph with dummy children U_i of the nodes in $\mathcal{B}$. See Figure 7.11.

Next, triangulate the extended free graph and form a junction tree. The junction tree will provide separators, which can be used to determine cut sets. You look for sets of separators blocking the flow from frontier nodes to border nodes. You start in the leafs with U -nodes and move backwards. Consider the junction tree in Figure 7.12. The separators 10, 11, and 13 d-separate the U -nodes from the rest. They form $\mathcal{B}$. As 10 is a frontier node, you cannot block it with nodes further back in the junction tree. Going backwards from the clique (10, 13, 14) you meet the separator (10, 14), and you find the cut set (10, 11, 14). Going backwards from the separator (10, 14), you meet a clique with the frontier node D_9 , and therefore you hereafter have to include D_9 in the cut sets. The same happens when you go backwards from the separator 11. The the new cut sets $\{10; 11; 14\}$, and $\{10; 14; 17; 23; D_9\}$.

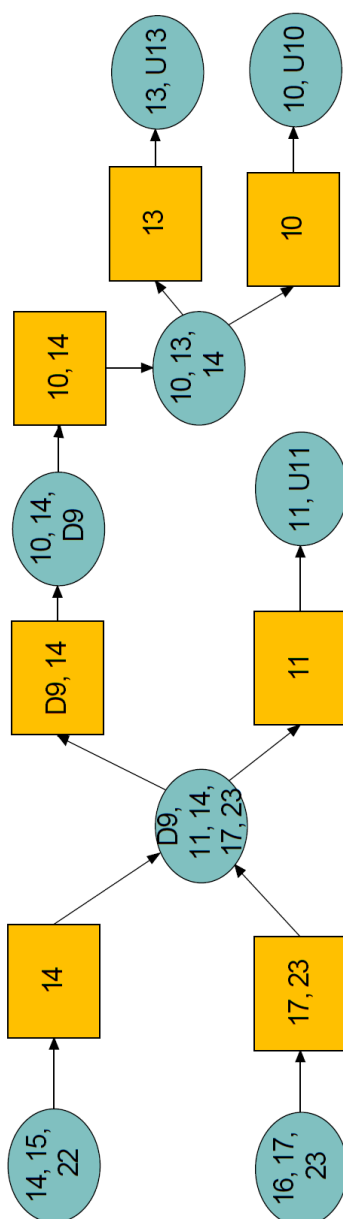


Figure 7.12: A junction tree for the extended free graph in Figure 7.11.

7.4.8 Irregular networks

If the network is irregular, the search for frontier nodes is more involved. An active path from $\mathcal{Inf}$ to $U \in \mathcal{U}$ ends with a directed series $V_k \rightarrow \dots \rightarrow U$. The first node in this series is a frontier node. For regular networks, the frontier consists of common ancestors of $\mathcal{B}$ and $\mathcal{Inf}$. For irregular networks we need to define $\mathcal{I}$ differently: for $X \rightarrow Y \rightarrow Z$ with $Y \in \mathcal{I}$ and with $Z \in \mathcal{P}$ or with a descendant in $\mathcal{P}$ we also include X in $\mathcal{I}$.

7.4.9 An iterative procedure

You may choose the nodes in the cut set iteratively, and whenever a node has been selected, you may renew the analysis. In the example for this paper it is certain that node 10 always will be part of the domain for δ_{10} . Hence, we need not look for ways of blocking information coming from 10, and the ancestors of 10 are only relevant if they are ancestors of other information nodes. Actually, for the ID in Figure 7.4, inclusion of 10 does not change the analysis because the only parent of 10 is D_9 , and it is an information node.

7.4.10 Conclusion and future work

We have established methods for finding approximate representations of future decisions policies through information enhancement. The methods do not determine all possible candidates for information enhancement. First of all, a good cut set does not necessarily contain only nodes between the border and the frontier. That is, you may go behind the frontier. Furthermore, we have only treated approximation of the last decision. Usually, the decision in question has several future decisions to consider, and you may look for a combined approximation of several future decisions.

Chapter 8

Conclusion

In many application domains we are interested in probability distribution over time of events. In addition, observations are performed irregularly along time, due to either a limitation of the measurement process or lack of a constant immanent rate. Thus, we are interested in a general efficient method to model, learn and reason about structured stochastic processes that produce qualitative and quantitative data at regularly or irregularly spaced time-points. Markov models that may be used to support these tasks differ in the way they represent time. Discrete-time Markov models are not well suited for irregular time settings: Hidden Markov Models, and Dynamic Bayesian Networks (DBNs) in general require the specification of a time granularity between each two consecutive observations. This requirement leads to computationally inefficient learning and inference when the modeled time granularity is finer than the time spent between consecutive observations, and to an information loss in the opposite case. In both cases, inference is limited to multiples of the modeled time granularity and would otherwise require the learning of a new model. Markov models that represent time continuously, on the other hand, handle well time irregularity. The framework of Continuous Time Bayesian Network explicitly represents temporal dynamics and allows for a factorization of the state space of a process, resulting in better modeling and inference features. Such models are based on graph structures, grounded on the theory of Bayesian Networks (BN). In this dissertation an overview of the probabilistic graph-

ical models for continuous time inference has been proposed. In particular, the framework of Continuous Time Bayesian Network has been described in both its semantic aspects and regarding the inference task. In this dissertation an overview of the probabilistic graphical models for continuous time inference has been proposed, the class of processes representable by CTBNs has been investigated. Many of the available inference algorithms for CTBNs proposed in literature have been studied and included in the CTBN toolbox. We then exploited the CTBN framework to diagnose the cause of cardiogenic heart failure and to anticipate its likely evolution. The proposed model overcomes the strong modeling and computational limitations of Discrete-time Markov models. The first clinical application of the recent developments in the research area of probabilistic graphical modeling in continuous time have been described. This approach allows a direct representation of time and offers a valid computational machinery for medical inference. The predictions emerging from the proposed scenarios confirmed the impact of duration of pathological events in the diagnosis process. The key idea of embedding the decision of a clinician to choose and apply the correct therapy on a system like those proposed, gave the input for further investigation on probabilistic graphical models. In the field of structured decision problems, the influence diagram framework has been identified as the most appropriate in order to deal with sequential decisions that best reproduce the flow of information in a continuous time horizon. Finally, we reported a new approach regarding approximate representation of optimal strategies from Influence Diagrams. Where the solution algorithm for influence diagrams is too space and time consuming, and where the policy functions for the decisions have so large domains, an assumption has to be taken in order to reduce the information. We call the approach *information enhancement*, assuming that we reduce the future information scenarios by adding information links. A systematic way of determining information links to add has been proposed.

The limits of exponential distribution to model temporal dynamic systems suggest that further results could be obtain investigating the ways that phase-type distributions can be used to model complex durations. That includes more work examining the impact of phase distributions on learn-

ing and inference. Further investigations can be performed in the direction of the learning procedure. In particular quantitative parameters might be tuned to achieve posterior probabilities that better fit with expectations derived from knowledge. This could be achieved by learning the CIMs directly from data. The relevance of the continuous time Bayesian network model could be further increased through the embedding of the CTBN model into a DSS which assists the decision maker to choose and apply the correct decision. In the clinical domain the decision could be the therapy.

Bibliography

[Hug, 2010] (2010). <http://www.hugin.com>.

[Gen, 2010] (2010). <http://genie.sis.pitt.edu>.

[Aalen, 1975] Aalen, O. (1975). Statistical inference for a family of counting processes. *Doctoral dissertation, Department of Statistics, University of California, Berkeley*.

[Allen, 1984] Allen, J. (1984). Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

[Allison, 2004] Allison, P. (2004). Event history analysis. *Handbook of data analysis*, pages 369–386.

[Andersen and Borgan, 1985] Andersen, P. and Borgan, Ø. (1985). Counting Process Models for Life History Data: A Review [with Discussion and Reply]. *Scandinavian Journal of Statistics*, 12(2):97–158.

[Andersen et al., 1993] Andersen, P., Borgan, Ø., Gill, R. D., and Keiding, N. (1993). *Statistical models based on counting processes*. New York: Springer Verlag.

[Andreassen et al., 1989] Andreassen, S., Jensen, F., Andersen, S., Falck, B., Kjærulff, U., Woldbye, M., Sørensen, A., Rosenfalck, A., and Jensen, F. (1989). MUNIN an expert EMG assistant. *Computer-aided electromyography and expert systems*, pages 255–277.

[Arroyo-Figueroa and Sucar, 1999] Arroyo-Figueroa, G. and Sucar, L. (1999). A temporal Bayesian network for diagnosis and prediction. In

- Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 13–22.
- [Augusto, 2001] Augusto, J. (2001). The logical approach to temporal reasoning. *Artificial Intelligence Review*, 16(4):301–333.
- [Augusto, 2005] Augusto, J. (2005). Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33(1):1–24.
- [Beek and Cohen, 1990] Beek, P. and Cohen, R. (1990). Exact and approximate reasoning about temporal relations. *Computational intelligence*, 6(3):132–144.
- [Besnard et al., 2002] Besnard, Y., Rialle, V., Vila, A., Reymond, F., Abaoub, L., and Dahou, M. (2002). NEUROP: an expert system in electromyography based on a multilevel knowledge representation. In *Engineering in Medicine and Biology Society, 1991. Vol. 13: 1991., Proceedings of the Annual International Conference of the IEEE*, pages 1302–1303. IEEE.
- [Bettini et al., 2000] Bettini, C., Jajodia, S., and Wang, S. (2000). *Time granularities in databases, data mining, and temporal reasoning*. Springer Verlag.
- [Bichindaritz and Conlon, 1996] Bichindaritz, I. and Conlon, E. (1996). Temporal knowledge representation and organization for case-based reasoning. In *Third International Workshop on Temporal Representation and Reasoning, 1996.(TIME'96), Proceedings.*, pages 152–159.
- [Blossfeld and Hamerle, 1992] Blossfeld, H. and Hamerle, A. (1992). Unobserved heterogeneity in event history models. *Quality and Quantity*, 26(2):157–168.
- [Blossfeld et al., 1989] Blossfeld, H., Hamerle, A., and Mayer, K. (1989). *Event history analysis: Statistical theory and application in the social sciences*. Lawrence Erlbaum.
- [Blossfeld and Rohwer, 2002] Blossfeld, H. and Rohwer, G. (2002). *Techniques of event history modeling: New approaches to causal analysis*. Lawrence Erlbaum.

- [Boudali and Dugan, 2006] Boudali, H. and Dugan, J. (2006). A continuous-time Bayesian network reliability modeling and analysis framework. *IEEE Transactions on Reliability*, 55(1):86–97.
- [Brophy et al., 2004] Brophy, J., Dagenais, G., McSherry, F., Williford, W., and Yusuf, S. (2004). A multivariate model for predicting mortality in patients with heart failure and systolic dysfunction. *The American journal of medicine*, 116(5):300–304.
- [Burstein and Holsapple, 2008] Burstein, F. and Holsapple, C. (2008). *Handbook on Decision Support Systems 2: Variations*. Springer-Verlag New York Inc.
- [Chittaro and Montanari, 2000] Chittaro, L. and Montanari, A. (2000). Temporal representation and reasoning in artificial intelligence: Issues and approaches. *Annals of Mathematics and Artificial Intelligence*, 28(1):47–106.
- [Combi et al., 2010] Combi, C., Keravnou-Papailiou, E., and Shahar, Y. (2010). *Temporal Information Systems in Medicine*. Springer.
- [Combi et al., 1995] Combi, C., Pincioli, F., and Pozzi, G. (1995). Managing different time granularities of clinical information by an interval-based temporal data model. *Methods of information in medicine*, 34(5):458–474.
- [Cooper, 1990] Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405.
- [Cowell, 1999] Cowell, R. (1999). *Probabilistic networks and expert systems*. Springer Verlag.
- [Das and Musen, 1994] Das, A. and Musen, M. (1994). A temporal query system for protocol-directed decision support. *Methods of Information in Medicine*, 33(4):358–370.
- [Dean and Kanazawa, 1989] Dean, T. and Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150.

- [Drake, 1962] Drake, A. (1962). Observation of a Markov process through a noisy channel. *Ph.D. thesis*.
- [Druzdzal and Suermondt, 1994] Druzdzal, M. and Suermondt, H. (1994). Relevance in probabilistic models: 'Backyards' in a 'small world'. In *Working notes of the AAAI-1994 Fall Symposium Series: Relevance*, volume 101.
- [Duffie et al., 1996] Duffie, D., Schroder, M., and Skidas, C. (1996). Recursive valuation of defaultable securities and the timing of resolution of uncertainty. *The Annals of Applied Probability*, 6(4):1075–1090.
- [El-Hay et al., 2006] El-Hay, T., Friedman, N., Koller, D., and Kupferman, R. (2006). Continuous time markov networks. In *Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence*.
- [El-Hay et al., 2008] El-Hay, T., Friedman, N., and Kupferman, R. (2008). Gibbs sampling in factorized continuous-time Markov processes. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 169–178.
- [Fan and Shelton, 2009] Fan, Y. and Shelton, C. (2009). Learning continuous-time social network dynamics. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 161–168. AUAI Press.
- [Fan and Shelton, 2008] Fan, Y. and Shelton, C. R. (2008). Sampling for Approximate Inference in Continuous Time Bayesian Networks. In *Tenth International Symposium on Artificial Intelligence and Mathematics*.
- [Fan et al., 2010] Fan, Y., Xu, J., and Shelton, C. (2010). Importance Sampling for Continuous Time Bayesian Networks. *Journal of Machine Learning Research*, 11:2115–2140.
- [Fernandez et al., 2005] Fernandez, J., Martinez-Selles, M., and Arredondo, M. (2005). Bayesian networks and influence diagrams as valid decision support tools in systolic heart failure management. In *Computers in Cardiology, 2004*, pages 181–184. IEEE.

- [Fuglsang-Frederiksen et al., 1989] Fuglsang-Frederiksen, A., Rønager, J., and Vingtoft, S. (1989). PC-KANDID: An expert system for electromyography. *Artificial Intelligence in Medicine*, 1(3):117–124.
- [Gallardo et al., 1987] Gallardo, R., Gallardo, M., Nodarse, A., Luis, S., Estrada, R., Garcia, L., and Padron, O. (1987). Artificial Intelligence in the electromyographic diagnosis of cervical roots and brachial plexus lesions. *Electroencephalography and Clinical Neurophysiology*, 66.
- [Garg et al., 2005] Garg, A., Adhikari, N., McDonald, H., Rosas-Arellano, M., Devereaux, P., Beyene, J., Sam, J., and Haynes, R. (2005). Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *The Journal of the American Medical Association*, 293(10):1223–1238.
- [Gatti et al., 2011] Gatti, E., Luciani, D., and Stella, F. (2011). A continuous Time Bayesian Network model for cardiogenic heart failure. *Special Issue on Health Care Management; Flexible Services and Manufacturing Journal*. Submitted paper.
- [Gopalratnam et al., 2005] Gopalratnam, K., Kautz, H., and Weld, D. (2005). Extending continuous time bayesian networks. In *AAAI’05: Proceedings of the 20th national conference on Artificial intelligence*, volume 20, pages 981–986.
- [Hayes and McCarthy, 1969] Hayes, P. and McCarthy, J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502.
- [Heckerman et al., 1992] Heckerman, D., Horvitz, E., and Nathwani, B. (1992). Toward normative expert systems: Part I. The Pathfinder project. *Methods of information in medicine*, 31(2):90–105.
- [Henke, 1994] Henke, A. (1994). Scheduling Space Shuttle Missions:using object-oriented techniques. *AI expert*, 9:16–16.
- [Herbrich et al., 2007] Herbrich, R., Graepel, T., and Murphy, B. (2007). Structure from failure. In *Proceedings of the 2nd USENIX workshop on Tack-*

- ling computer systems problems with machine learning techniques, pages 1–6. USENIX Association.
- [Hobbs, 1985] Hobbs, J. (1985). Granularity. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*.
- [Hope et al., 2007] Hope, L., Nicholson, A., Korb, K., and Clayton, V. (2007). TakeheartII: A tool to support clinical cardiovascular risk assessment. Technical report, Clayton School of Information Technology, Monash University.
- [Howard and Matheson, 1981] Howard, R. and Matheson, J. (1981). Influence diagrams. *The Principles and Applications of Decision Analysis*, 2:721–762.
- [Hulst, 2006] Hulst, J. (2006). Modeling physiological processes with dynamic Bayesian networks. *Master thesis, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Pittsburgh*.
- [ICTBIO, 2008] ICTBIO (2008). European Commission Ict-bio 2008: Conference report.
- [Jamieson, 1990] Jamieson, P. (1990). Computerized interpretation of electromyographic data. *Electroencephalography and Clinical Neurophysiology*, 75(5):392–400.
- [Jensen, 2008] Jensen, F. (2008). Approximate representation of optimal strategies from influence diagrams. In *Proceedings of the 4th European Workshop on Probabilistic Graphical Models*.
- [Jensen and Gatti, 2010] Jensen, F. and Gatti, E. (2010). Information enhancement for approximate representation of optimal strategies from influence diagrams. In *Proceedings of the 5th European Workshop on Probabilistic Graphical Models*.
- [Jensen et al., 1994] Jensen, F., Jensen, F., Dittmer, S., et al. (1994). From influence diagrams to junction trees. In *Proceedings of the tenth conference on Uncertainty in Artificial Intelligence*.

- [Jensen and Nielsen, 2007] Jensen, F. and Nielsen, T. (2007). *Bayesian networks and decision graphs*. Springer Verlag.
- [Kahn et al., 1991] Kahn, M., Fagan, L., and Tu, S. (1991). Extensions to the time-oriented database model to support temporal reasoning in medical expert systems. *Methods of Information in Medicine*, 30(1):4–14.
- [Kahn and Marrs, 1995] Kahn, M. and Marrs, K. (1995). Creating temporal abstractions in three clinical information systems. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 392–396. American Medical Informatics Association.
- [Kan and Shelton, 2008] Kan, K. F. and Shelton, C. R. (2008). Solving Structured Continuous-Time Markov Decision Processes. In *Tenth International Symposium on Artificial Intelligence and Mathematics*.
- [Keravnou and Washbrook, 1990] Keravnou, E. and Washbrook, J. (1990). A temporal reasoning framework used in the diagnosis of skeletal dysplasias. *Artificial Intelligence in Medicine*, 2(5):239–265.
- [Kjaerulff, 1995] Kjaerulff, U. (1995). dHugin: A computational system for dynamic time-sliced Bayesian networks. *International journal of forecasting*, 11(1):89–111.
- [Kjaerulff and Madsen, 2007] Kjaerulff, U. and Madsen, A. (2007). *Bayesian networks and influence diagrams: a guide to construction and analysis*. Springer Verlag.
- [Kohane, 1987] Kohane, I. (1987). Temporal reasoning in medical expert systems. Technical report, Boston Univ., MA (USA).
- [Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.
- [Lando, 1998] Lando, D. (1998). On Cox processes and credit risky securities. *Review of Derivatives research*, 2(2):99–120.

- [Lauritzen and Nilsson, 2001] Lauritzen, S. and Nilsson, D. (2001). Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251.
- [Lauritzen and Spiegelhalter, 1988] Lauritzen, S. and Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224.
- [Levy et al., 2006] Levy, W., Mozaffarian, D., Linker, D., Sutradhar, S., Anker, S., Cropp, A., Anand, I., Maggioni, A., Burton, P., Sullivan, M., et al. (2006). The Seattle Heart Failure Model: prediction of survival in heart failure. *Circulation*, 113(11):1424–1433.
- [Lipscombe, 1989] Lipscombe, B. (1989). Expert systems and computer-controlled decision making in medicine. *AI & Society*, 3(3):184–197.
- [Lipsky, 1992] Lipsky, L. (1992). *Queueing theory: a linear algebraic approach*. New York: Macmillan Publishing Company.
- [Mascherini and Stefanini, 2010] Mascherini, M. and Stefanini, F. (2010). Using weak prior information on structures to learn Bayesian Networks. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 413–420. Springer.
- [McDermott, 1982] McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive science*, 6(2):101–155.
- [Middleton et al., 1991] Middleton, B., Shwe, M., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H., and Cooper, G. (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Medicine*, 30:241–255.
- [Miller et al., 1976] Miller, A., Merkhofer, M., Howard, R., Matheson, J., and Rice, T. (1976). Development of automated computer aids for decision analysis. Technical report, 3309, SRI International, Menlo Park, Calif.

- [Minka, 2001] Minka, T. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.
- [Moler and Van Loan, 2003] Moler, C. and Van Loan, C. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49.
- [Montanari and Pernici, 1993] Montanari, A. and Pernici, B. (1993). Temporal reasoning. *Temporal Databases: Theory, Design and Implementation, Database Systems and Applications Series*, pages 534–562.
- [Murphy, 2002] Murphy, K. (2002). Dynamic bayesian networks: representation, inference and learning. *Doctoral dissertation, University of California, Berkley*.
- [Nathwani et al., 1990] Nathwani, B., Heckerman, D., Horvitz, E., and Lincoln, T. (1990). Integrated expert systems and videodisc in surgical pathology: an overview. *Human pathology*, 21(1):11–27.
- [Ng et al., 2005] Ng, B., Pfeffer, A., and Dearden, R. (2005). Continuous time particle filtering. In *International Joint Conference on Artificial Intelligence*, volume 19, page 1360.
- [Nielsen, 2001] Nielsen, T. (2001). Decomposition of influence diagrams. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 144–155.
- [Nielsen and Jensen, 2004] Nielsen, T. and Jensen, F. (2004). Advances in decision graphs. *Studies in fuzziness and soft computing*, 146:137–160.
- [Nodelman, 2007] Nodelman, U. (2007). Continuous Time Bayesian Networks. *Doctoral dissertation, Stanford University*.
- [Nodelman and Horvitz, 2003] Nodelman, U. and Horvitz, E. (2003). Continuous time Bayesian networks for inferring users’ presence and activities with extensions for modeling and evaluation. Technical report, Microsoft Research.

- [Nodelman et al., 2002] Nodelman, U., Shelton, C., and Koller, D. (2002). Continuous time bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 378–387.
- [Nodelman et al., 2003] Nodelman, U., Shelton, C., and Koller, D. (2003). Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence*, pages 451–458.
- [Nodelman et al., 2005] Nodelman, U., Shelton, C., and Koller, D. (2005). Expectation maximization and complex duration distributions for continuous time Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence*, volume 5, pages 421–430.
- [Norris, 1997] Norris, J. (1997). *Markov chains*. Cambridge University Press.
- [Ozsoyoglu and Snodgrass, 1995] Ozsoyoglu, G. and Snodgrass, R. (1995). Temporal and real-time databases: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 513–532.
- [Pani et al., 2001] Pani, G. et al. (2001). Temporal representation and reasoning in artificial intelligence: A review. *Mathematical and Computer Modelling*, 34(1-2):55–80.
- [Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.
- [Pinto and Reiter, 1995] Pinto, J. and Reiter, R. (1995). Reasoning about time in the situation calculus. *Annals of Mathematics and Artificial Intelligence*, 14(2):251–268.
- [Provan and Ball, 1983] Provan, J. and Ball, M. (1983). The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12:777–788.
- [Rose et al., 2005] Rose, C., Smaili, C., and Charpillet, F. (2005). A dynamic Bayesian network for handling uncertainty in a decision support system adapted to the monitoring of patients treated by hemodialysis. In *Tools*

- with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pages 5–598. IEEE.
- [Ross, 2007] Ross, S. (2007). *Introduction to probability models*. Academic Press.
- [Shachter, 1986] Shachter, R. (1986). Evaluating influence diagrams. *Operations Research*, 34(6):871–882.
- [Shafer and Shenoy, 1988] Shafer, G. and Shenoy, P. (1988). Local computation in hypertrees. Technical report, School of Business, University of Kansas.
- [Shenoy, 1992] Shenoy, P. (1992). Valuation-based systems for Bayesian decision analysis. *Operations research*, 40(3):463–484.
- [Sim et al., 2001] Sim, I., Gorman, P., Greenes, R., Haynes, R., Kaplan, B., Lehmann, H., and Tang, P. (2001). Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 8(6):527–534.
- [Stillman et al., 1993] Stillman, J., Arthur, R., and Deitsch, A. (1993). Tachyon: A constraint-based temporal reasoning model and its implementation. *ACM SIGART Bulletin*, 4(3):1–4.
- [Tawfik and Neufeld, 1994] Tawfik, A. and Neufeld, E. (1994). Temporal bayesian networks. In *Proceedings of the TIME-94–International Workshop on Temporal Representation and Reasoning*.
- [Tawfik and Neufeld, 2000] Tawfik, A. and Neufeld, E. (2000). Temporal reasoning and Bayesian networks. *Computational Intelligence*, 16(3):349–377.
- [Twardy et al., 2006] Twardy, C., Nicholson, A., Korb, K., and McNeil, J. (2006). Epidemiological data mining of cardiovascular Bayesian networks. *electronic Journal of Health Informatics*, 1(1):e3.

- [van Gerven et al., 2008] van Gerven, M., Taal, B., and Lucas, P. (2008). Dynamic Bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41(4):515–529.
- [Vila et al., 1985] Vila, A., Ziebelin, D., and Reymond, F. (1985). Experimental EMG expert system as an aid in diagnosis. *Electroencephalography and Clinical Neurophysiology*, 61(3):S240.
- [Vilain et al., 1989] Vilain, M., Kautz, H., and van Beek, P. (1989). Constraint propagation algorithms for temporal reasoning: a revised report. In *Readings in qualitative reasoning about physical systems*, pages 373–381. Morgan Kaufmann Publishers Inc.
- [Xiang et al., 1992] Xiang, Y., Eisen, A., MacNeil, M., and Beddoes, M. (1992). Quality control in nerve conduction studies with coupled knowledge-based system approach. *Muscle & nerve*, 15(2):180–187.
- [Xiang et al., 1993] Xiang, Y., Pant, B., Eisen, A., Beddoes, M., and Poole, D. (1993). Multiply sectioned Bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine*, 5(4):293–314.
- [Xu and Shelton, 2008] Xu, J. and Shelton, C. (2008). Continuous time Bayesian networks for host level network intrusion detection. *Machine Learning and Knowledge Discovery in Databases*, pages 613–627.
- [Yamaguchi, 1991] Yamaguchi, K. (1991). *Event history analysis*. Sage Publications, Inc.