TEMPORAL REASONING IN MEDICINE USING DYNAMIC BAYESIAN NETWORKS

by

Senthil Kumar Nachimuthu

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Biomedical Informatics

The University of Utah

May 2012

Copyright © Senthil Kumar Nachimuthu 2012

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Senthil Kumar Nachimuthu**

has been approved by the following supervisory committee members:

Peter J. Haug	, Chair	**1/17/2012** Date Approved
Bruce Bray	, Member	**1/12/2012** Date Approved
Brad R. Farr	, Member	**1/24/2012** Date Approved
J. Brent Muhlestein	, Member	**1/23/2012** Date Approved
Scott P. Narus	, Member	**1/12/2012** Date Approved

and by **Joyce A. Mitchell** , Chair of the Department of **Biomedical Informatics**

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

Temporal reasoning denotes the modeling of causal relationships between different variables across different instances of time, and the prediction of future events or the explanation of past events. Temporal reasoning helps in modeling and understanding interactions between human pathophysiological processes, and in predicting future outcomes such as response to treatment or complications. Dynamic Bayesian Networks (DBN) support modeling changes in patients' condition over time due to both diseases and treatments, using probabilistic relationships between different clinical variables, both within and across different points in time.

We describe temporal reasoning and representation in general and DBN in particular, with special attention to DBN parameter learning and inference. We also describe temporal data preparation (aggregation, consolidation, and abstraction) techniques that are applicable to medical data that were used in our research. We describe and evaluate various data discretization methods that are applicable to medical data. Projeny, an open-source probabilistic temporal reasoning toolkit developed as part of this research, is also described.

We apply these methods, techniques, and algorithms to two disease processes modeled as Dynamic Bayesian Networks. The first test case is hyperglycemia due to severe illness in patients treated in the Intensive Care Unit (ICU). We model the patients' serum glucose and insulin drip rates using Dynamic Bayesian Networks, and recommend insulin drip rates to maintain the patients' serum glucose within a normal range. The model's safety and efficacy are proven by comparing it to the current gold standard. The second test case is the early prediction of sepsis in the emergency department. Sepsis is an acute life threatening condition that requires timely diagnosis and treatment. We present various DBN models and data preparation techniques that detect sepsis with very high accuracy within two hours after the patients' admission to the emergency department.

We also discuss factors affecting the computational tractability of the models and

appropriate optimization techniques. In this dissertation, we present a guide to temporal reasoning, evaluation of various data preparation, discretization, learning and inference methods, proofs using two test cases using real clinical data, an open-source toolkit, and recommend methods and techniques for temporal reasoning in medicine.

To my family,
my teachers, and
the giants of human knowledge
on whose shoulders I stand.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

$PaCO_2$ Arterial Partial Pressure of Carbon Dioxide

ANN Artificial Neural Network

AUC Area Under the (ROC) Curve

BNJ Bayesian Network tools in Java (a software)

BNT Bayes Net Toolbox (a Matlab toolbox software)

BP Blood Pressure

CDR Clinical Data Repository

CDS Clinical Decision Support

CPT Conditional Probability Table

DAG Directed Acyclic Graph

DBN Dynamic Bayesian Network

DBP Disatolic Blood Pressure

ECG Electrocardiogram

ED Emergency Department

EHR Electronic Health Record

EM Expectation Maximization (algorithm)

EMR Electronic Medical Record

HDD Healthcare Data Dictionary

HELP Help Evaluation through Logical Processing

HL7 Health Level Seven

HMM Hidden Markov Model

HR Heart Rate

ICU Intensive Care Unit

KFM Kalman Filter Model

LIMID Limited Memory Influence Diagram

LOINC Logical Observation Identifier Names and Codes

MAP Mean Arterial Pressure

MDL Minimum Description Length (algorithm)

MDP Markov Decision Process

MODS Multiple Organ Dysfunction Syndrome

MPD Marginal Probability Distribution

MPE Most Probable Estimate

MRN Medical Record Number

NPV Negative Predictive Value

POMDP Partially Observable Markov Decision Process

PPV Positive Predictive Value

PTXT Pointer to Text

ROC Receiver Operator Characteristic (curve)

RR Respiratory Rate

SBP Systolic Blood Pressure

SIRS Systemic Inflammatory Response Syndrome

STRICU Shock Trauma Respiratory Intensive Care Unit

ACKNOWLEDGMENTS

A serendipitous discovery of the field of medical informatics a decade ago in an internet cafe in Chennai set me on a quest to learn formally about what I had been doing haphazardly on my own. A decade later, I look back in gratitude at all the people who have helped and supported me along the way. First and foremost, I thank my parents for patiently supporting me while I've been so far away from them for so long. My wife Myungae and my relatives have also given me the strength and confidence to pursue my dream of pursuing research in this field. My family stood by me during the ups, and more so during the downs. I couldn't have done this without their support.

I want to thank my mathematics tutor, Mr. K. Prathaban, who helped to broaden my perspective about learning and life during my formative years. My high school physics teacher, Mrs. Sudha Venkatesan, stimulated my curiousity for learning by experimentation. My uncles, Dr. M. Kuppusamy and Dr. K. Kaliappan, opened my eyes to the beauty of science, and to the art and science of medicine. Dr. C. Sivalingam encouraged me by lending me his computer and letting me work and play with his homegrown patient information system, and my aunt K. Vimala amplified my interest in computer programming. It is with this background that I gleaned that a systematic method to capture, use and analyze clinical information and knowledge can help medical care tremendously. A visit to the medical records room of my alma mater, Stanley Medical College, to find the records of my former patients among decades worth of medical records convinced me that using technology to handle information is the best way to practice evidence-based medicine and to discover new clinical knowledge. My surgery professors, Dr. R. Surendran and Dr. G. Balakrishnan, encouraged me to pursue this field. I wouldn't have realized the joy of learning and the fruits of perseverance without their help and support. Stanley Medical College, its hospitals and fellow Stanleans taught me a lot about life by infecting me with enthusiam and immunizing me with resilience. I hope to forever carry the Stanlean Spirit forward.

When I arrived in a new land determined to work on clinical decision support, Prof. Peter Haug provided me the guidance and support to learn about this field. He encouraged me to test my limits and was patient with my mistakes. I thank him for immersing me in the world of clinical decision support. My committee members, Drs. Bruce Bray, Brad Farr, Scott Narus, Brent Muhlestein, and Adam Wilcox, provided insights from a variety of backgrounds and perspectives, including cardiology, computer science and machine learning. Dr. Lee Min Lau and Shaun Shakib helped me learn a lot about terminologies and ontologies, which helped me with collecting, organizing and using clinical data for my research. Dr. Jau-Huei Lin, Vikrant Deshmukh, and Anthony Wong helped to formulate and refine my research questions. Drs. Kathy Sward and Jason Jones helped me with the data sets for my research. Many researchers, clinicians, and staff of Intermountain Healthcare helped with my research at LDS Hospital and Intermountain Medical Center. My friend Dr. Binu Mathew provided input into my research from a computer science perspective. I thank them all for their invaluable help.

The Department of Neurobiology and Anatomy, Health Sciences Center Information Technology Services, the Division of Epidemiology, and the Department of Biomedical Informatics at the University of Utah, and 3M Health Information Systems, Inc., provided me financial support during various phases of my graduate program. I also received travel awards from the University of Utah Graduate School. I thank these organizations for their generous support, and I thank my mentors at these organizations, Dr. Kurt Albertine, Cheri Hunter, Dr. Lee Min Lau, and Dr. Matthew Samore, for helping me learn through real-world experience.

My friends Abdy, Aldo, Arun, Binu, Divya, Raj, Raja, Sameer, Suren, and Vikrant provided me moral support that helped me to persevere whenever I hit an obstacle. I also received help from authors of research articles and open source software. Dr. Kevin Murphy deserves a special mention for his articles and toolkits.

I sincerely think that all the hidden and observed nodes, recognized and unrecognized edges within and across different time slices, and their stochastic and deterministic influences in this large temporal probabilistic network of my past have made my present state possible. So long, and thanks for all the fun!

CHAPTER 1

INTRODUCTION

> There are, in general, two ways to predict the future. You can, for example, use horoscopes, tea leaves, tarot cards, a crystal ball, and so forth. Collectively, these are known as "nutty methods." Or you can put well-researched facts into sophisticated computer models, more commonly referred to as "a complete waste of time." While all these approaches have their advantages, I find it's a lot easier and more economical to simply make stuff up.
>
> - Scott Adams, 2008[1].

The quote above from Scott Adams humorously sums up the difficulty in building computational models to predict the future, especially in a complex and highly variable field such as medicine. This dissertation is an attempt at making the prediction of the future or the explanation of the past or the present using probabilistic methods in certain clinical conditions not a complete waste of time, and on the contrary, computationally tractable, repeatable and sufficiently accurate to be useful.

Time is an essential component of the art and science of medicine. Cause-and-effect relationships between diseases and clinical features, and between interventions and clinical features, are interpreted in the context of the time by which the cause precedes the effect. These relationships have different timescales, some producing an effect in the order of seconds, with others producing an effect in the order of decades. Clinical pathophysiology and treatment involves many of these overlapping cause-and-effect processes. An understanding of their complex interactions in terms of causality and temporal nature is essential to interpret the patients' clinical conditions, treat them effectively, and assess their response to treatments[2].

Temporal reasoning involves the understanding of relationships between different variables across time. It can contribute to models that explain past events and predict future events based on available data. An ability to explain the past can provide an understanding of a patient's disease processes and response to treatment. Predicting future events

is a natural continuation of the temporal modeling process, and can help in estimating the prognosis and planning the treatment. Prediction can also help to avert impending complications, both long-term (e.g., graft survival), and short-term (e.g., anticoagulation therapy). Some clinical conditions such as long-term graft survival involve pathophysiological processes that span several years. On the other hand, the pathophysiological processes involved in insulin release and glucose homeostasis span a short interval in the order of a few hours[3].

Temporal modeling and prediction have been used routinely in several fields such as financial projections and weather forecasting. However, temporal reasoning techniques have not been studied to such an extent in clinical medicine. This is due to various unique characteristics such as the complexity of medical science, practice of medicine, medical data, the inherent uncertainty underlying the medical decision making process, and the unavailability of tools to accommodate these unique characteristics. This dissertation is an attempt to define these unique characteristics, accommodate and overcome the challenges they pose, develop a tool to facilitate probabilistic temporal reasoning in medicine, and apply these tools and techniques to two very different temporal problems in clinical medicine to prove their use, validity, and generalizability. This dissertation also aims to be a primer on probabilistic temporal reasoning in medicine.

1.1 The Problem

The objective of clinical treatment is to change the clinical condition of a patient from a less healthy to a more healthy state. Predicting the evolution of the clinical condition and of future events is a natural part of this process. This process is of special interest in patients' clinical conditions that change very rapidly over time due to critical illnesses and aggressive treatments. Heuristic prediction of future events by the human clinical expert is an uncertain process that is not well understood. This process also exhibits high variability both across different human experts, and across different instances in the same expert[4, 5, 6]. A repeatable, formal, evidence-based model becomes highly desirable to improve understanding and accuracy, and to reduce uncertainty and variability of these predictions.

Many techniques for temporal learning and prediction have been used with varying levels of success. These techniques span a spectrum of methods with purely rule-based models at one end and complex mathematical pattern recognition models at the other, and various hybrid models in between. The accuracy and the acceptability of these models for clinical use also vary. Probabilistic models are one of the well-understood mathematical models used in various fields including biomedicine. Probabilistic models have been used extensively in the biomedical domain to predict diseases and outcomes based on clinical signs, symptoms, and laboratory findings. Naive Bayesian and Bayesian Network techniques have proven to be highly accurate. These techniques have traditionally been used with single-point-in-time models, such as predicting the possibility of various diseases from vital signs, chest X-ray[7, 8, 9, 10], coronary angiography[11], and laboratory findings[12, 13, 14, 15]. These models have not been used to build or test models that explicitly track the change in patients' conditions over time. Temporal models often prove to be more complex than atemporal models[16]. The lack of temporal probabilistic models have been attributed to various problems such as the lack of powerful and easy to use modeling tools, unavailability of temporal data in an easily computable format, the large amount of missing data in temporal data sets, the intractability of temporal learning algorithms, and problems in defining temporal relationships[17]. Clinical data are collected with the primary objective of providing clinical care by human experts (physicians, nurses, and others), and are often not collected with the objective of electronic processing and reasoning. Even in well-designed electronic medical record systems, the data have many unique characteristics that make the application of machine learning techniques very difficult. These peculiarities have to deal with varying granularities of time, the enormity of missing data in electronic medical records, and the large variations of structure and format of the same data[18].

Even though probabilistic temporal reasoning techniques have been used with much success in other domains, they have a limited proof of successful use in the clinical domain. Hidden Markov Models (HMM) is one of the well-studied temporal reasoning techniques used in biomedicine[19, 20]. HMMs help to model a Markov process as a temporal model where an unobservable variable transitions between different states over time, and is measured by means of another proxy variable known as the observed variable.

The Markov Property reduces the computational complexity by allowing the conditional probabilities for state transmission and observation to be constant over time given a set of preconditions, as explained in Chapter 2. However, HMMs do not lend well to modeling very complex processes. Dynamic Bayesian Networks (DBNs) are a general case of HMMs which support modeling more complex processes than is possible with HMMs[21]. DBNs have been used in other domains to model complex temporal processes. However, DBNs have not been used as extensively as HMMs in the biomedical domain except in a handful of cases due to various challenges introduced above and explained in detail in subsequent chapters.

In addition to probabilistic techniques, nonprobabilistic techniques have also been used for temporal reasoning in various domains including biomedicine. These techniques include pattern recognition techniques such as Artificial Neural Networks (ANNs), and statistical techniques such as logistic regression[22, 23, 24, 25]. However, both pattern recognition and statistical techniques can only recognize correlation, and lack the expressivity of probabilistic models to denote complex, hierarchical, or temporal causal relationships[26, 27, 28]. These techniques also have predetermined input (independent), and output (dependent) variables. These techniques lack the flexibility of probabilistic models, which allow any variable whose value is not known to be treated as an output variable. ANNs also work as black boxes, and hence, their predictions are considered to lack proper explanations, which are required in the practice of evidence-based medicine. Hence, the pattern recognition and statistical techniques have found limited acceptance except in very specific problems with a single output variable such as prediction of prognosis following the diagnosis of a disease or a treatment procedure. These techniques and their advantages and limitations are explained further in Chapter 2.

1.2 A Solution Using Dynamic Bayesian Networks

Dynamic Bayesian Networks theoretically provide a very expressive and flexible model to solve temporal problems in medicine. However, this involves various challenges due both to the nature of the clinical domain, and the nature of the DBN modeling and inference process itself. The challenges from the clinical domain include an insufficient knowledge of temporal interactions of pathophysiological processes in the medical literature,

the sparse nature and variability of medical data collection, and the difficulty in preparing and abstracting clinical data in a suitable format without losing valuable information in the process. Challenges pertaining to the DBN methodology and implementation include the lack of tools that allow easy modeling of temporal processes, lack of algorithms that support models involving various Markov processes with different orders, paucity of tools that support different cases in the data sets spanning different durations, difficulties with input and output of large amounts of structured data into and out of these tools, and the computational complexity and tractability[29][30].

Overcoming these challenges will help to solve various clinical temporal reasoning problems. This dissertation describes experiments involving temporal reasoning in medicine using Dynamic Bayesian Networks (DBN). The goal of the research was to evaluate various data preparation and temporal modeling methods, and apply them to test cases in clinical medicine to evaluate the hypothesis of whether accurate temporal reasoning is possible in the clinical domain using DBNs. Evaluation of the methods and techniques were done by comparing the predictions of the temporal models to available gold standards - the clinical decisions of the physicians and other clinical experts. The results of these two test cases prove the hypothesis that temporal reasoning using Dynamic Bayesian Results can be done in an accurate, clinically useful, computationally efficient, and timely manner. The details of the two test caes, their hypotheses, and the materials, methods, and results of the experiments are presented in Chapters 4 and 5.

The dissertation provides a review of various temporal reasoning techniques and related works[31]. This dissertation also describes an approach to probabilistic temporal reasoning in medicine using Dynamic Bayesian Networks. This approach is the result of various experiments performed by the author as part of the doctoral work underlying this dissertation. This dissertation describes various problems with clinical data, and explains techniques that are applicable to clinical data preparation for use in temporal modeling. Data preparation involves temporal abstraction and discretizing continuous variables into discrete states. Pitfalls with manual data discretization and the advantages of algorithmic data discretization are described through various experiments. Two different data discretization algorithms implemented in the Weka software platform were used in this research. Results obtained from models created using these algorithms are compared with

results obtained with manual data discretization, showing the advantage of the algorithmic data preparation process.

A toolkit that was developed to perform the experiments in this dissertation is also described. This toolkit, Projeny (acronym for Probabilistic Networks Generator in Java), was created to enable easy authoring of temporal models, and allows data binding, parameter learning, and inference through an easy-to-use graphical interface. Projeny is based on two other open source toolkits - BNJ (Bayesian Network Tools in Java), and JMatLink. Projeny provides a user interface to BNT (Bayes Net Toolbox), a toolkit that runs inside a Matlab environment. Projeny has been released as open source software and is being updated regularly. BNT provides all the DBN learning and inference algorithms for the experiments described in this dissertation. The advantages of BNT over other temporal modeling toolkits is also described.

The research also involved testing the techniques and tools to solve real clinical problems. Two test cases were used to test and prove the feasibility, accuracy, validity, and usefulness of these tools and techniques in modeling and solving clinical problems and improving the clinical care of patients. Both experiments were performed as pure data-only studies using the retrospective data of patients treated at Intermountain Healthcare in Salt Lake City, Utah, USA.

The first test case involves the modeling of glucose homeostasis of critically ill patients in the intensive care unit. Patients who are critically ill lose control of serum glucose levels even if they are not diabetic, leading to a high increase in serum glucose. Uncontrolled hyperglycemia in these patients has been proven by several authors to worsen their clinical condition and the outcomes, leading to higher mortality and morbidity[32]. On the contrary, aggressive glucose control in these patients has been shown to improve the outcomes and reduce the mortality and morbidity[32][33][34]. However, it must be noted that this claim has been disputed by some authors[35, 36]. Our hypothesis underlying this test case is that the DBN model can accurately predict the serum glucose levels and recommend insulin doses better than those of the current gold standard (eProtocol-insulin) to maintain a normal serum glucose level. eProtocol-insulin is a computerized rule-based protocol developed and used at Intermountain Healthcare to recommend insulin doses to maintain patients' serum glucose within normal levels[37]. The first experiment tests the accuracy

of prediction of serum glucose and insulin drip rates in temporal data of patients. The second experiment compares the insulin doses recommended by the DBN model to those produced by the current gold standard.

The second test case involves early prediction of sepsis in patients seen at the emergency departments (ED) of Intermountain Healthcare. Sepsis is a dangerous disease condition that evolves very rapidly, and early detection and treatment of sepsis leads to significantly improved outcomes as shown by several authors[38, 39, 40, 41]. The hypothesis underlying this experiment is that the DBN model can predict the presence of sepsis in ED patients accurately and in a timely manner compared to the clinician's diagnosis which is the current gold standard. The objective of this experiment is to predict sepsis in patients before their blood culture and sensitivity results are available so that treatment can be initiated sooner. The experiments measure the accuracy of predicting whether a patient has sepsis using the patients' 3-hour, 6-hour, 12-hour, and 24-hour data since admission.

The dissertation also describes the computational complexity of the models and the techniques that are useful to reduce the space and time complexity of these models. These are described in context while describing the data preparation techniques and the experiments involving the two test cases. The dissertation provides an insight into various techniques that can be used to reduce the computational complexity of these models and to make them computationally tractable.

1.3 Outline of the Dissertation

This dissertation as a whole describes research involving the application of DBNs in the clinical domain, challenges and techniques for temporal data preparations, various challenges involving temporal reasoning in general and those specific to the clinical domain, results from applying these techniques to two clinical conditions, and a temporal reasoning toolkit created as part of this research. This dissertation concludes with the limitations of the techniques in this research as applicable to the biomedical domain, with pointers for future research to overcome these limitations.

Chapter 2 provides a comprehensive overview of probabilistic and other temporal reasoning techniques. It also disambiguates between temporal reasoning and temporal

representation, as the former term is used by several authors to denote the latter since the two are closely related. Chapter 2 also describes relevant works involving temporal reasoning in biomedical and other domains. It also provides an overview of various toolkits available for probabilistic temporal reasoning.

Chapter 3 describes the nature of clinical data that makes temporal reasoning difficult. It presents various techniques used in this research that are appropriate for data preparation and abstraction for temporal reasoning while preserving the information content. It provides a roadmap to other researchers involved in temporal reasoning for data preparation. This chapter also describes the Projeny toolkit developed as part of this research project, along with the other toolkits it depends and is based on, namely, BNT[42], JMatLink[43], and BNJ[44]. It also describes a roadmap for future development of the Projeny toolkit as an open source software project.

Chapters 4 and 5 describe the two test cases. Chapter 4 describes the modeling and experiments involving the first test case, glucose homeostasis in the ICU. It describes the materials and methods, modeling challenges, and discusses the results of various experiments involving this test case. Chapter 5 describes the second test case, early prediction of sepsis in the emergency department. This chapter likewise describes the materials and methods, challenges with data preparation and modeling, and discusses the results. These two experiments involve very different diseases and modeling and prediction problems. These two chapters prove the validity and accuracy of the methods and toolkits produced as part of the author's doctoral research, and support the claim of their generalizability to temporal modeling problems in the clinical domain at large.

Chapter 6 discusses the validity of results obtained from both experiments for their utility in clinical care. It also discusses the validity of these results in the context of interpolation and extrapolation of the range of values of clinical variables encountered in the study. It discusses the generalizability and external validity of the methods as well. Chapter 6 concludes the dissertation by discussing the lessons learned, the original contributions to the field of temporal reasoning in medicine, and outlines the future research directions. This chapter also describes the limitations of DBNs and recommends more complex techniques that may be capable of overcoming these limitations. This may serve as a topic to be explored by future research projects.

CHAPTER 2

REVIEW OF THEORETICAL FOUNDATIONS AND RELATED WORKS

This chapter begins by defining and disambiguating the terms temporal reasoning and temporal representation in the context of this dissertation. This chapter then looks at the various attributes of time, and provides a description of temporal representation from the perspective of temporal ontology and logic. Temporal logic and constraints based on first-order logic are defined. This discussion forms the basis for understanding the complex challenges in modeling time and answering questions pertaining to temporal logic. This chapter then discusses various temporal databases that support the storage of temporal attributes and querying of temporal relationships between various data elements. Temporal databases and temporal query languages can encapsulate manual data aggregation tasks involved in data preparation as well as provide data storage for temporal reasoning toolkits[45], and are hence briefly described in this dissertation.

This chapter then provides a nonexhaustive overview of various temporal reasoning methods that are applicable to biomedicine. Various statistical, probabilistic, and pattern-recognition-based temporal modeling and prediction techniques are discussed, with special attention to probabilistic techniques. These techniques abstract many of the temporal constraint solving problems. An overview of the temporal logic and constraint solving problems, though not central to this dissertation, is warranted to set the stage for the need for higher levels of abstractions and to describe how these abstractions satisfy the temporal constraints.

This chapter also outlines challenges faced in probabilistic temporal reasoning. These challenges are due to various factors. Some of the challenges are inherent in temporal representation and reasoning; other challenges are due to the nature of the medical science, due to imprecise understanding of causal and temporal relationships; still other challenges

are due to the nature of the data captured in the practice of medicine, with the primary goal being the provision of clinical care and not computational modeling.

This chapter provides a review of relevant works in temporal reasoning in biomedicine. The advantages of methods and techniques described in this dissertation over these works will be described in later sections. A description of the Projeny temporal modeling toolkit developed as part of this research is presented on the Projeny website[**?**].

2.1 Temporal Representation and Reasoning Defined

Temporal representation denotes the formal methods used to structure and express time in a logical and computable manner. This is done with a variety of formal constructs to denote time and temporal relationships between events.

Temporal reasoning denotes the modeling of causal or explanatory relationships between different variables or events, and the ability to predict future events or to explain past events. However, some authors use the term 'temporal reasoning' to denote what is described as 'temporal representation' in this dissertation. This ambiguity becomes especially difficult to distinguish in the context of temporal logic as described below.

2.2 Temporal Representation

A computable model of time is critical for problems in the biomedical domain such as diagnosis, prognosis, prediction of future events, explanation of past medical events, recognition of trends, continuous monitoring and control, spatiotemporal modeling of epidemics, and so on. The need for a computable model of time has also been well recognized in areas such as robotics, intelligent agents, planning of future actions, financial analysis and projections, modeling of weather and climate, etc. Many formalisms for modeling time have been developed by various authors. Special database constructs and temporal query languages have also been developed to store and query temporal data. Temporal logic can then be applied to the data to answer questions of a temporal nature or to detect any inconsistencies in the temporal data.

2.2.1 Various Attributes of Time

Time can be modeled in a variety of ways depending on the domain and the nature of the available data. Some of the more common formalisms are described below.

2.2.1.1 Absolute and Relative Time

Time can be defined as a process that happens on its own, with various events attached to points of time at which they occur and periods of time over which they hold. This absolute notion of time stems from the idea that time is an independent dimension over which things exist and events occur. A relative notion of time discounts the importance of time by itself, and describes the relevance of time by only expressing the universe of interest to be made of various events that are temporally related[46, 47]. An example of the former would be a model of time in the universe, whereas an example of the latter would be the human perception of time in terms of events that are observed. From a computational modeling perspective, both models of time are possible, and have been described by various authors. Temporal representation systems have been built that implement either of these models or a combination of the two, and temporal reasoning techniques that accommodate either of these models are available[48].

2.2.1.2 Implicit and Explicit Time

A medical record may provide the time during which an event happened either implicitly by relating it to other events, or explicitly by specifying the time when an event occurred. The medical record may say ‘the patient had chest pain after running uphill’ or ‘the patient had chest pain around 7PM on Monday evening’. Implicit time models require further analysis before they can be fit into a temporal model of events. Storing clinical data in temporal databases becomes especially harder in cases of discontinuous or disjoint interval data such as ‘the patient had chest pain on and off for several years while he was smoking and after he quit’. Words such as ‘before’, ‘after’, or ‘during’ are used to capture the temporal events in relation to one another, rather than capture the calendar date and time at or during which these events occurred. Hence, implicit and explicit time models are closely related to relative and absolute time models, respectively. Explicit time models capture time on its own, rather than modeling it in terms of events that happen

in the system. Explicit time models may be point-based, interval-based, or based on a combination of the two.

2.2.1.3 Discrete and Continuous Time

A temporal model may also be modeled as a sequence of discrete events that happen at points of time, or as a sequence of processes that occur continuously and overlap or meet one another[48]. For example, a patient's heart rhythm can change from normal sinus rhythm to ventricular fibrillation, provided the granularity of time is in the order of minutes. On the other hand, the serum troponin levels of a patient recovering from an acute myocardial infarction changes continuously over time when measured with the same granularity of time. A continuous time model is also known as a dense time model[49]. Both Situation Calculus and Event Calculus models described in Section 2.2.2 can model discrete changes well, but they do not lend well to modeling continuous changes. With these models, it is not possible to accurately describe a variable of interest at a point of time between two discrete events or measurements, though certain methods to represent continuous events with Situation Calculus[50], and Event Calculus[51] have been described. Some physiologic models based on mathematical equations try to model such continuous processes[15]. Several authors have described a combination of logic-based models with mathematical equation-based models[52]. Though probabilistic models such as Dynamic Bayesian Networks mathematically can be proven to work with varying granularities of time, currently available learning and inference algorithms require time to be modeled as a discrete quantity to perform tractable learning and inference with large DBN models[21, 53, 54].

2.2.1.4 Bounded and Unbounded Time

Bounded and unbounded temporal models denote whether the time-axis is finite or infinite. Almost all temporal reasoning problems in biomedicine assume a finite model of time. They attempt to model the temporal system and to make predictions or explanations over a finite period of time in the future or the past. However, tractable learning and inference is possible in probabilistic temporal networks with sequences of unbounded lengths by applying parameter tying through detection of equivalence classes [55].

2.2.2 Temporal Logic Ontologies and Formalisms

Among the many formalisms for representing temporal interactions in a system, Situation Calculus and Event Calculus are the ones most widely studied and adopted. Techniques such as Situation Calculus and Event Calculus allow reasoning with implicit time models, and to an extent, explicit time models. Both are first-order logic formalisms that are used to express the state, actions, and changes in a system. They are also useful to solve temporal constraint satisfaction problems and to predict the future state of a system or to explain the past state.

Situation Calculus (SC) models the system in terms of *situations*, which describe the entire state of the universe at a given instance of time, and various *actions* that are possible given the situation. This first-order-based logic was first described by McCarthy and Hayes[56]. Both possible scenarios (e.g., the patient's serum glucose is 50 mg/dl, encountered in hypoglycemic patients), and hypothetical situations (e.g., the patient's serum glucose is 2mg/dl, which is not seen in live human beings) are considered, even though the state of the universe (e.g., neurological and cardiac functions of the patient) is not completely defined in hypothetical situations. A set of *fluents* provide known information about a given situation, whether possible or hypothetical. For example, we may have one fluent *glucose(s)* that expresses '*the patient's serum glucose during situation s*', and another fluent that provides composite knowledge such as '*the patient cannot be alive in a situation s when the serum glucose is 2 mg/dl*'. A set of all allowed *actions* along with the corresponding resulting situations are defined for the given situation. A knowledge of all possible situations, fluents, and actions is adequate to describe the state of the given system and all possible outcomes by using first-order logic [57, 56, 58].

Event Calculus (EC) is a logic-programming framework introduced by Kowalski and Sergot[59]. Event Calculus describes a temporal system in terms of *events* and *properties*. Properties are fluents that hold true between the events that initiate and terminate them. Event Calculus provides a simpler model of a system by only modeling known events and properties without the knowledge of the entire universe of interest. Various events and properties can be represented along a time-axis as multiple parallel and overlapping timelines. These timelines all start and end at specific events, and hold various properties throughout the timeline. The timelines may be continuous or disjoint, and maintain causal

temporal relationships between them, with cause preceding the effect.

In contrast, Situation Calculus aims to model all possible permutations of the universe of interest and transitions between them. On a time-axis, Situation Calculus can be represented as a multidimensional branching and merging model. From the above discussion, it can be seen that Situation Calculus lends itself better to relational or implicit time models, and Event Calculus lends well to both implicit (relative), and explicit (absolute) time models.

In addition to Situation Calculus and Event Calculus, other authors have described formalisms such as *features and fluents* formalism (Sandewall)[60], *fluent calculus* (Hölldobler and Thielscher)[61, 62], and others[63, 64].

2.2.3 Temporal Constraint Representation and Satisfaction

Various forms and qualitative and quantitative formalisms are available to represent temporal constraints. Reasoning about temporal constraint involves whether a set of variables (objects, fluents, states, etc.) exist to satisfy a given constraint, and choosing the most probable temporal relationship between these variables (objects, fluents, etc.). The most popular qualitative formalisms to represent temporal constraints are Interval Algebra (Allen)[48], Point Algebra (Vilain and Kautz)[65], and others[66]. These formalisms denote temporal constraints using relationships such as 'before', 'after', 'during', 'at', and so on. Many algorithms are available for performing reasoning using these constraint representations[67, 68, 69], and a detailed survey of these algorithms is presented in [70].

Allen's Interval Algebra uses thirteen different relationships to denote temporal constraints between two intervals. These relationships are *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, along with the inverses of these six relationships, and the relationship *equal*[48]. These are illustrated in Figure 2.1. For example, the interval inflammation may *start* the interval pain, and inflammation may have a *before-inverse* relationship to infection.

Vilain and Kautz's Point Algebra denotes temporal constraints using points in time rather than intervals of time. Point Algebra uses disjunctions of three basic relationships, $<$ (before), $=$ (at the same time), and $>$ (after). From these three, seven vectors are then formed $>$, $<$, $=$, $>=$, $<=$, $<>$ (not at the same time), and $<=>$ (not known).

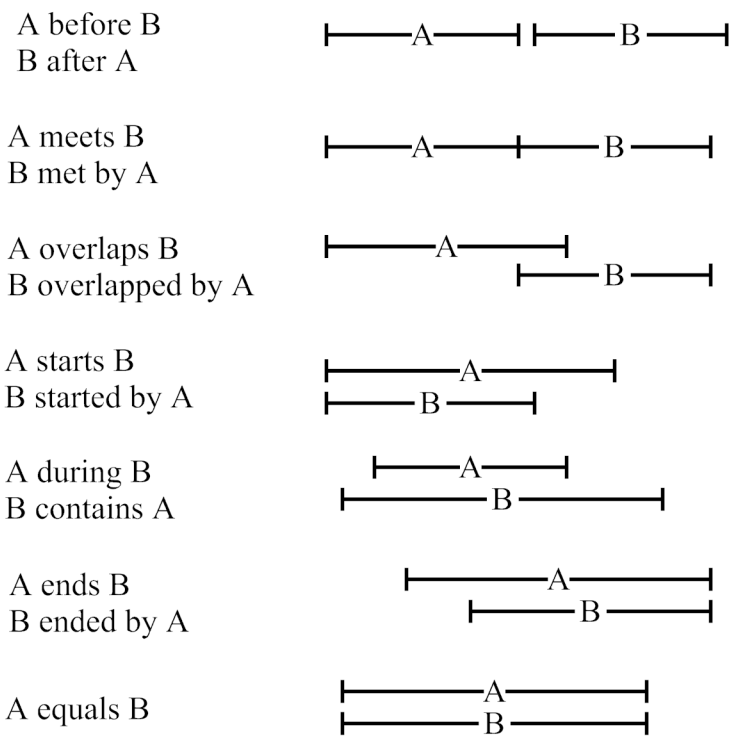


Figure 2.1: Allen's Interval Algebra

Addition and multiplication logic between these vectors are then defined. The truth tables for addition and multiplication for the vectors used in point algebra are then described[65]. Addition is used to combine two different measures of relationships between two points. Multiplication is used to find the relationships between two points, given their separate relationships with a third point. For example, addition will be used to find the relationship between infection and sepsis, if we know that infection always precedes sepsis, and infection does not occur simultaneously as sepsis. If we know that infection precedes sepsis, and sepsis precedes severe sepsis, we can use multiplication to determine that infection precedes severe sepsis. However, vector algebra does not answer all possible combinations of vector additions and multiplications[65]. For example, if we know that infection precedes inflammation, and infection precedes pain, this information alone is insufficient to answer the temporal relationship between inflammation and pain.

Quantitative formalisms for temporal constraint representation have also been described[71, 72]. A well-known quantitative algorithm is the Distance Algebra (Dechter, Mieri, and Pearl)[71]. Distance Algebra denotes the values of various entities over time using both

unary and binary constraints, to capture both point- and interval-based temporal data. A unary constraint denotes the value of an entity at a specific point in time. A binary constraint denotes the value of the entity between two points in time. Interactions of various constraints have been described for Distance Algebra. The reader is referred to [71] for a detailed description. Combinations of qualitative and quantitative temporal constraint representation and solving have also been described[73, 74, 75]. A detailed overview of temporal logic and constraint formalisms are presented by Chittaro and Montanari[76].

2.2.4 Temporal Databases and Query Languages

Data storage and retrieval mechanisms become necessary to query temporal data easily for applying temporal logic-based operations on them. Many temporal databases are enhancements to well-known database technologies such as file-based, hierarchical, or relational database systems. The temporal query languages are often extensions to the Structural Query Language (SQL) to support temporal queries.

Database formalisms to support temporal data use point-based or interval-based ontologies of time to capture temporal semantic relationships in clinical data. The earliest example of a temporal database is the Time Oriented Database (TOD) by Wiederhold et al.[77] TOD is a file-based database system designed to represent time using timestamps for medical data in medical records. Another early system that used timestamps for clinical data are the Rx system by Blum[78]. Kahn et al. describe Extended Temporal Network (ETNET), an object-oriented database system that captures temporal relationships for the ONCOCIN decision support system[79]. Pinciroli et al. describe some limitations of relational databases in representing temporal data, and propose an object-oriented database model to support temporal data representation[80].

The well-known examples of temporal query languages are extensions to the Structured Query Language (SQL) to support queries with a temporal nature. SQL does not provide a simple way to formulate temporal queries, and requires complex date-time comparisons to manually create temporal queries[81]. Hence, various authors proposed extensions to the SQL standard to support temporal queries. Temporal query formalisms support queries such as ‘retrieve events that happened between when the patient had chest pain and was admitted to the emergency department’, ‘retrieve events since the patient was

diagnosed with diabetes mellitus', and so on. Das et al. describe Chronus[82], and other extensions to SQL to support temporal queries, and their effectiveness to support clinical decision support[83, 84, 82]. Combi et al. describe GCH-OSQL and S-WATCH-QL, two extensions to SQL to support temporal queries in clinical applications[85, 86]. Snodgrass et al. describe TSQL2, a feature-rich extension to SQL to support temporal queries[87]. The draft version of the upcoming ISO SQL3 standard (the newer version of the current SQL standard) included temporal extensions to the SQL known as SQL/Temporal, but the extension has since been withdrawn[81, 88].

2.3 Overview of Temporal Reasoning

As described in Section 2.1, the term 'Temporal Reasoning' is used in this dissertation to denote the modeling of causal relationships across time, and inferring the state of a system at any given point in time, including prediction of future events or explanation of past events. Temporal reasoning is an essential part of diagnosis, treatment, and prognosis in clinical medicine. The task of temporal reasoning is performed by human experts as an implicit part of the clinical care delivery process. Evidence-based medicine (EBM) constructs such as clinical protocols and guidelines also implicitly use temporal reasoning as part of a supervised decision support process. Clinical decision making is inherently uncertain, due to difficulties in measuring the system under consideration, uncertainties in eliciting these measurements, the large number of interacting disease processes and clinical manifestations, and due to an incomplete understanding of the interactions as well as the individual disease processes themselves. Temporal reasoning becomes harder given these circumstances, compared to well-studied but complex domains where these techniques have been applied with a higher degree of success, such as continuous speech recognition, and autonomous agent modeling.

The understanding and prediction of clinical events by human experts becomes difficult for rapidly changing systems such as serum glucose control and insulin dosing in the ICU, especially when the data are incomplete. Computerized temporal reasoning becomes desirable in these cases. Many methods of temporal reasoning have been attempted in clinical medicine. The accuracy of temporal reasoning depends both on accurate modeling and meaningful data preparation, as shown in Chapter 3. In this chapter, we describe

various modeling techniques that are applicable to temporal reasoning in the biomedical domain.

2.3.1 Temporal Reasoning Techniques

Both qualitative and quantitative methods have been studied to solve temporal reasoning problems. Qualitative methods include rule-based, frame-based, and logic-based methods[89, 90]. Qualitative methods are closely related to logical formalisms of time described in Sections 2.2.2, and 2.2.3.

Quantitative Temporal reasoning techniques vary from the simple difference equations models, to differential equation-based models, regression models, to the more complex probabilistic models such as Bayesian models, and pattern recognition models such as Artificial Neural Networks. The terms 'time series analysis', and 'time series prediction' are more popular with regression-based and similar models used in financial analysis and econometrics. Detailed descriptions of mathematical models for temporal analysis and prediction are provided by Box et al.[91], Brockwell and Davis[92], Hamilton[93], and Kedem and Fokianos[94]. These works describe regression-based models including auto-regressive moving average (ARMA), logistic regression, Kalman filter, and similar methods for models involving continuous as well as nominal and ordinal discrete variables. Support Vector Machines (SVM) have also been used in temporal reasoning in recent times[95, 96, 97].

Artificial Neural Networks also prove to applicable for solving temporal reasoning and prediction problems[98, 99, 100]. They have been applied in several domains including financial analysis, agent modeling, and biomedicine. However, due to the increasing emphasis on evidence-based medicine, Artificial Neural Networks are not as well accepted in medicine as regression or probabilistic models because of their 'black box' nature in their inability to explain their reasoning[101, 102, 28].

Probabilistic methods such as Bayesian analysis have traditionally been used for static models, or models whose variables have a static value. Hidden Markov Models and Dynamic Bayesian models are becoming increasingly popular to solve temporal problems in the biomedical domain. A description of these methods is provided in Section 2.4.

2.4 Probabilistic Methods in Temporal Reasoning

Probabilistic methods used in temporal reasoning include Dynamic Bayesian Networks (DBN), Hidden Markov Models (HMM), Kalman Filters, Partially Observed Markov Decision Processes (POMDP), and Limited Memory Influence Diagrams (LIMID). All the latter models can be described as specializations of Dynamic Bayesian Networks. All of these models use the Bayes theorem and the Markov property to model conditional probabilities both within a given instance of time as well as over a period of time. Hence, we briefly describe Bayesian probabilities and the Markov property before describing the probabilistic models mentioned above.

2.4.1 Bayes Theorem

Bayes theorem describes the conditional probabilities of different stochastic random variables in a probabilistic model. A stochastic variable is one that takes a value from a stochastic or probabilistic space, and its value is expressed as a probability distribution over a set or range of possible values, in contrast with a deterministic variable whose value can be known exactly.

A Naive Bayesian model is the simplest instantiation of the Bayes theorem. A Naive Bayesian model assumes strong conditional independence between various independent variables that are related to a single dependent variable. For example, cough and fever can be modeled as independent variables that predict pneumonia. In a Naive Bayesian model, the three variables, cough, fever, and elevated white blood cell (WBC) counts are assumed to be conditionally independent of each other, and the possibility of other diagnoses is not considered unless they are mutually exclusive and exhaustive.

In a Multimembership Bayesian model, different clinical features can be modeled to be associated with different diseases, and each disease is considered to be independent of other diseases. However, this may not often be the case with clinical models, and there are often interactions between different groups of clinical features and underlying illnesses. Consideration of these interactions are often captured in the clinical environment as differential diagnoses, given the history and clinical condition of the patient's illness. Bayesian Networks (BN) overcome the limitations due to conditional independence assumptions in the Multimembership Bayesian models, and allow for complex probabilistic interactions

between various nodes in the model.

In its simplest form, often used in a Naive Bayesian model, the Bayes theorem relates the conditional and marginal probabilities of two variables as

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \tag{2.1}$$

where B has a nonzero probability.

For ease of explanation, we rewrite the equation 2.1in terms of a disease and its clinical finding. We use D to denote the disease and F to denote the finding as shown in equation 2.2.

$$P(D|F) = \frac{P(F|D)\ P(D)}{P(F)} \tag{2.2}$$

In equation 2.2, $P(D)$ represents the probability of disease in the general population, also known as prevalence or prior probability or pretest probability of the disease. $P(F|D)$ denotes the probability of a patient having the clinical finding when the patient has the disease. $P(F)$ represents the probability of the finding in the general population. $P(D|F)$ denotes the probability of a patient having a disease when the finding is present, also known as posterior probability of the disease. From equation 2.2, we can see that the posterior probability of a disease can be calculated from its prior probability as well as the evidence.

The law of total probability, also known as the expansion rule, states that

$$P(A) = \sum_{B} P(A|B)\ P(B) \tag{2.3}$$

Applying the law of total probability to the denominator in equation 2.2, we can express the probability of the finding as

$$P(F) = P(D)\ P(F|D) + P(\overline{D})\ P(F|\overline{D}) \tag{2.4}$$

where $\overline{D}$, and $\overline{F}$ are the complementary events of the disease and finding, often pronounced as ‘not D’, and ‘not F’, respectively. $\overline{D}$ denotes the absence of disease, and $\overline{F}$

denotes the absence of the finding. Equation 2.4 helps us to calculate the prior probability of the finding from the observed variables on the right-hand side of the equation.

By combining equations 2.2 and 2.4, we get

$$P(D|F) = \frac{P(F|D)\ P(D)}{P(D)\ P(F|D) + P(\overline{D})\ P(F|\overline{D})} \tag{2.5}$$

The chain rule of probability describes the calculation of the joint probability distribution of multiple variables using their conditional probabilities. The rule states that

$$P(X_1 = x_i,\ \ldots,\ X_n = x_n)\ =\ \prod_{i=1}^{n} P(X_i = x_i \mid X_{i-1} = x_{i-1},\ \ldots,\ X_1 = x_1) \tag{2.6}$$

which can be written in simple terms as

$$P(A,\ B,\ C,\ D)\ =\ P(A \mid B, C, D)\ P(B \mid C, D)\ P(C \mid D)\ P(D) \tag{2.7}$$

Combining the chain rule (equation 2.6) with the law of total probability (equation 2.3), we can calculate the probability of a disease given multiple findings. The joint probability assumes that different symptoms of a single disease are conditionally independent. In reality, many symptoms of a given disease are correlated. For example, in case of sepsis, white blood cell (WBC) count and body temperature are correlated. However, Bayes theorem proves to be fairly accurate for small violations of conditional independence.

2.4.2 BN Structure, Conditional Independence and d-separation

A Bayesian Network structure is expressed as a directed acyclic graph (DAG) where the nodes represent the random variables in the model, and the edges represent the conditional dependencies between the random variables. If a directed edge connects node A to node B, then node A is known as a parent of node B, and node B is known as a child of node A. The random variables may be continuous or discrete. The various categories of values of discrete variables are known as their states. The acyclic nature of the graph allows us to decompose the joint probability distribution into its constituent conditional probabilities by applying the chain rule. It is not possible to perform this decomposition using the chain

rule if the graph contains cycles. The order of decomposition is based on the conditional interdependencies described in the graph structure. The process of decomposing the joint probability distribution into its constituents is known as factorization. The ability to factorize a DAG makes it easier to calculate the joint probability distribution. The junction tree algorithm takes advantage of this property by converting a highly connected graph into a tree structure of cliques (known as a 'junction tree'), and then by performing exact inference on the junction tree[103].

We use the Bronchitis and Lung Cancer network, which is a subset of the popular Asia network described by Lautitzen and Spiegelhalter[103], to describe Bayesian Network structure and various properties of these models. The Asia network has more random variables than the Bronchitis and Lung Cancer network shown in Figure 2.2. The Bronchitis and Lung Cancer network illustrated in Figure 2.2 is used to explain the principles of Bayesian conditional probability and d-separation. Smoking, bronchitis, lung cancer, dyspnea, and chest X-ray findings are represented by the symbols S, B, L, D, and C, respectively.

A patient may develop bronchitis or lung cancer due to smoking or due to other causes. The probability of bronchitis and lung cancer in the general population are denoted by $P(B)$, and $P(L)$, the prevalence or prior or pretest probabilities of these two diseases. The probability of lung cancer given the knowledge of the patient's smoking habit is denoted by $P(L|S)$, and the probability of bronchitis when we know whether the patient smokes is denoted by $P(B|S)$.

The two diseases, bronchitis and lung cancer, may cause dyspnea (shortness of breath), and radiographic findings detected in a chest X-ray. The conditional probabilities of these two clinical findings (dyspnea and chest X-ray findings) given the knowledge of each of these two diseases (bronchitis and lung cancer) are described by the symbols $P(D|B, L)$, and $P(C|B, L)$, respectively. The conditional probabilities of the diseases given the patient's smoking habit, and those of the clinical features given the diseases, may be calculated from a large sample of cases and controls. For ease of explanation, all the variables in the above model are assumed to be binary - they are either present or absent. Chest X-ray findings are assumed to be either normal or abnormal, for ease of explanation.

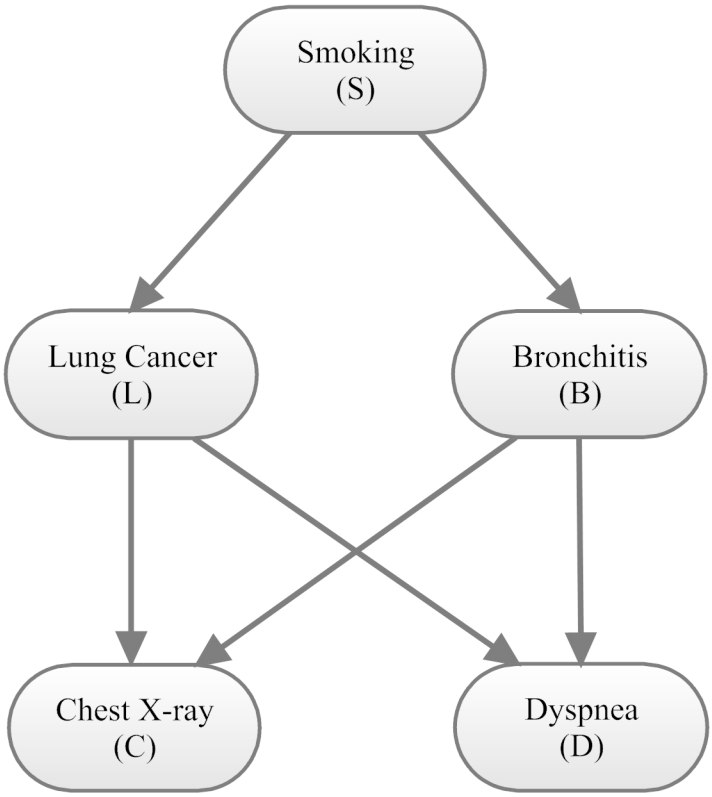


Figure 2.2: Bronchitis and lung cancer - a simple Bayesian Network

If there were n variables in the model, then the joint probability distribution will require an order of 2^n probabilities. This is computationally expensive, and an optimization method is required. It should be noted that the figure 2^n does not assume independence between the variables. Let us consider a subgraph of Figure 2.2 which consists of the three nodes lung cancer (L), dyspnea (D), and chest X-ray abnormalities (C). The joint probability of these three nodes is expressed as

$$P(L, D, C) = P(L)\, P(D, C|L) \tag{2.8}$$

If we assume that dyspnea and chest X-ray abnormalities are conditionally independent, then equation 2.8 can be rewritten as

$$P(L, D, C) = P(L)\, P(D|L)\, P(C|L) \tag{2.9}$$

Hence, we see that conditional independence reduces the number of terms in the joint probability distribution from $O(2^n)$ to $O(n)$. It must be noted that the edges in a Bayesian Network do not define a strict cause-effect relationship. They may explain relationships that are causal, logical, temporal, or conceptual[103]. Nodes that are connected directly to each other are necessarily conditionally dependent. However, nodes that are connected indirectly may or may not be conditionally independent. The principle of d-separation provides the necessary and sufficient conditions for conditional independence in nodes that are connected indirectly.

Consider the three graphs in Figure 2.3. The figure shows three graphs in converging, diverging, and sequential configurations. Two nodes in a graph are conditionally independent if and only if they are d-separated. Two nodes A and C in a graph are d-separated, if and only if there is a node B between them such that:

- the connection is sequential or diverging, and the intermediate node B is known.
- the connection is converging, and neither B nor any descendant of B is known.

While we note that conditional independence reduces the computational complexity of Bayesian Networks, and that d-separation defines conditional independence, an interesting phenomenon becomes apparent in cases of converging relationships. The principle of d-separation in the case of converging nodes can be described in terms of 'explaining away'. In graph 1 in Figure 2.3, we note that variables A and C try to explain the variable B. If variable B is known, then variables A and C share the explanation for B, and hence become conditionally dependent. For example, if dyspnea can be caused by both lung cancer and bronchitis, if we know whether the patient has dyspnea or not, the probability of lung cancer decreases as the probability of bronchitis increases and vice versa, even if we know that lung cancer and bronchitis are independent of each other. If one were to consider the relationship between these two diseases in the context of dyspnea, it would lead one to think that these diseases have an inverse relationship between them. This spurious conditional dependence and apparent selection bias is known in statistics as Berkson's paradox[104]. Hence, the author of the Bayesian Network must be aware of this paradox of 'explaining away' while designing a graph with converging edges.

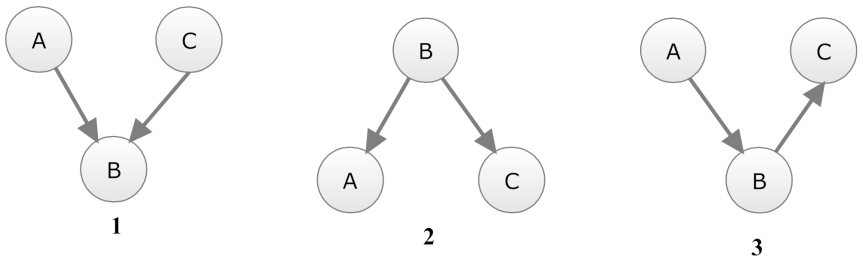


Figure 2.3: d-separation in Directed Acyclic Graphs

2.4.3 Markov Property

A Markov process is a stochastic process where the future state of a random variable does not depend on its past state if its present state is known. It may be noted that this property is similar to d-separation in case of sequential nodes, as described in Section 2.4.2, and Figure 2.3. An example is shown in Figure 2.4. A process which only depends on its immediate previous state is known as a first-order Markov process; one whose present state depends on its k previous states is known as a k^{th}order Markov process. Hence, a Markov process is a memoryless or a short-memory stochastic process. A Markov process that has a finite set of states is often known as a Markov chain. A Markov process may model time as a discrete or continuous quantity. Discrete time Markov processes are better defined, more tractable, and more popular than continuous time Markov processes. Each time instance in a discrete time Markov process is known as a timeslice.

Figure 2.4 shows sepsis as a finite state discrete Markov process. In this figure, sepsis has three states, namely no sepsis, sepsis, and septic shock. Each of the arrows in the figure represents a transition from one state to another or to itself. Each transition is associated with a transition probability. The figure represents sepsis as a first-order Markov process.

A stochastic process is a Markov process of order k if its present state is independent of all but its immediately previous k states. In other words, for a first-order Markov process,

$$P(X_{t+h} = y \mid X_s = x_s, \forall s \leq t) = P(X_{t+h} = y \mid X_t = x_t, \forall h > 0), \qquad \forall t, h > 0. \tag{2.10}$$

Similarly, for a k^{th} order Markov process,

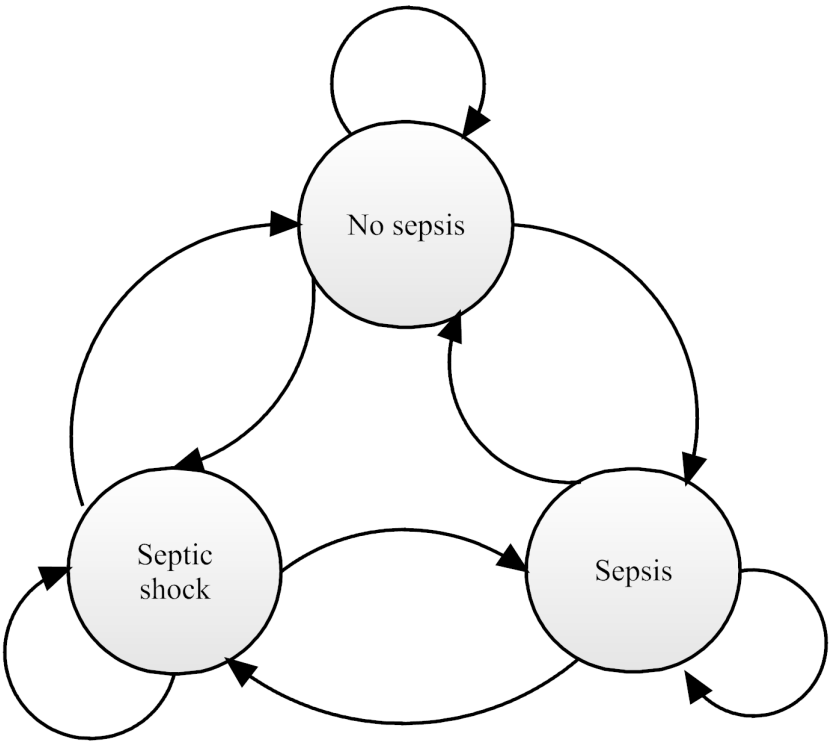


Figure 2.4: Sepsis as a Markov process

$$P(X_{t+h} = y \mid X_s = x_s, \forall s \leq t) = P(X_{t+h} = y \mid X_{t-m} = x_{t-m}, \forall m \leq k), \qquad \forall t, h > 0. \tag{2.11}$$

In equations 2.10, and 2.11, the quantity $P(X_{t+h} = y \mid X_t = x_t)$ is known as the transition probability. A Markov process may be time-homogeneous or time-nonhomogeneous. These may be simply known as homogeneous and nonhomogeneous, respectively. For a homogeneous Markov process, the transition probability for a given stochastic variable remains constant for all values of t. The transition probability of a nonhomogeneous Markov random variable changes with time. In other words, for a homogeneous Markov random variable,

$$P(X_{t+h} = y \mid X_t = x) = P(X_h = y \mid X_0 = x), \qquad \forall t, h > 0. \tag{2.12}$$

The equality assertion in equation 2.12 is not true for nonhomogenous Markov processes. Many biological processes are modeled as Markov processes, and Markov pro-

cesses have received good acceptance in modeling biomedical systems for predicting prognoses and outcomes. Modeling biological processes as homogenous Markov processes allows parameter tying and a reduction in the number of parameters required to describe the joint probability for a Markov process with a large time duration. Modeling temporal biological processes as homogenous Markov processes provides accurate estimations even for nonhomogenous Markov processes, as shown in Chapters 4 and 5.

2.4.4 Hidden Markov Models

A Hidden Markov Model (HMM) is the simplest type of a Dynamic Bayesian Network (DBN) with one hidden node and one observed node. A Hidden Markov Model represents the stochastic process in terms of a hidden variable and an observed variable. In a Hidden Markov Model, the hidden node is represented as a discrete variable, whereas the observed variable may be discrete or continuous. The hidden variable cannot be measured directly. It is measured through a proxy variable known as the observed variable. The observed variable is related to the hidden variable through an emission (or observation) probability. The state transitions of the hidden variable are described by the transition probability, which denotes the Markov nature of the stochastic process. The probabilistic relationships between various states of sepsis (no sepsis, sepsis, septic shock), and WBC count (low, normal, high) are shown in Figure 2.5 as a Hidden Markov Model. Transition probabilities are denoted by a_{pq} and observation probabilities are denoted by b_{pq}.

In Figures 2.5, and 2.6, sepsis is represented as a Hidden Markov Model. Squares denote discrete variables and circles denote continuous variables. Shaded nodes are observed and clear nodes are hidden. Figure 2.5 denotes the state transitions, and various transition and observation probabilities using a single timeslice, whereas, Figure 2.6 represents the same Hidden Markov Model using three timeslices. Furthermore, Figure 2.5 denotes the observed nodes as discrete nodes, whereas Figure 2.6 denotes the observed nodes as continuous nodes. The transition probability is denoted by a, and the observation probability is denoted by b. The prior probabilities of the directed acyclic graph are denoted by π. The tuple (π, a, b) denotes the parameters of the Hidden Markov Model.

Hidden Markov Models can be used to predict temporal as well as atemporal sequences. Atemporal sequences that lend well to Hidden Markov Models include gene

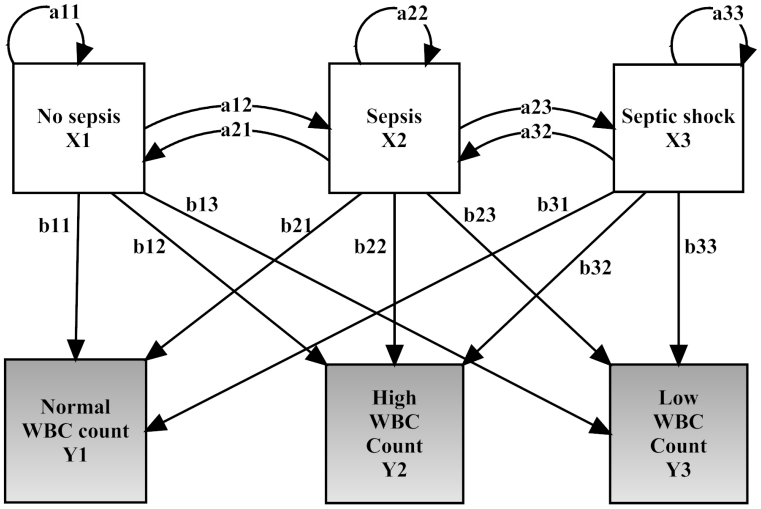


Figure 2.5: Sepsis state transitions as a Hidden Markov Model

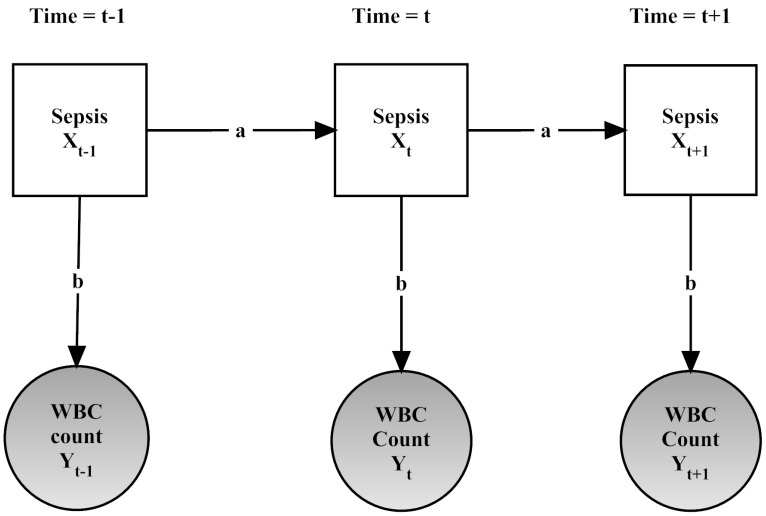


Figure 2.6: Sepsis Hidden Markov Model with 3 timeslices

and protein sequences. Hidden Markov Models are used extensively in speech recognition, machine translation, motion and gesture recognition, gene sequence prediction, and prediction of clinical outcomes and prognosis such as tumor recurrence after treatment and graft survival in transplant recipients. Complex Hidden Markov Models such as factorial and hierarchical Hidden Markov Models can be built by extending simple Hidden Markov Models. Rabiner provides a good description of Hidden Markov Models in [105].

2.4.5 Kalman Filter Models

Kalman Filter Models (KFM), also known as linear dynamical systems (LDS), or state-space models (SSM), model the hidden node as a continuous random variable, in contrast to a Hidden Markov Model. The transition and observation functions are assumed to be linear-Gaussian, and the system is assumed to be jointly Gaussian. Instead of conditional probability tables, the joint probability is expressed as a probability density function. Kalman Filters are used in problems such as tracking an aircraft or a projectile using a radar, and in tracking financial parameters, since the variables in these models are continuous in nature, with linear relationships between them. The linear-Gaussian nature of the probability functions necessitate that these functions be unimodal. However, this is not always the case in real-world clinical systems, such as predicting the outcome of a patient treated for a myocardial infarction. It is also not possible to model all the hidden clinical variables as continuous variables, or the probabilities as linear-Gaussian functions. Due to these reasons, Kalman Filter models have not found significant acceptance in the clinical domain except in signal processing domains such as electrocardiogram (ECG) or imaging. A detailed description of Kalman Filter Models from a Hidden Markov Model perspective is presented in [106], and [107].

2.4.6 Dynamic Bayesian Networks

Dynamic Bayesian Networks represent the state-space of a system in terms of multiple random variables with complex probabilistic interactions. The variables may change in value over time as seen in temporal models, or due to sequential information as seen in genetic or proteomic sequences. The term ‘dynamic’ denotes that the values of the variables change over time, and not that the model’s structure itself changes over time. In other words, the system being modeled is a dynamic one, in contrast with a static Bayesian Network where the variables have a fixed value.

A more appropriate term is ‘Temporal Bayesian Networks’, but the term ‘Dynamic Bayesian Networks’ has received wider acceptance and more popularity. The probabilistic interactions themselves follow Bayesian principles (Section 2.4.1), and the temporal probabilistic relationships follow Bayesian principles as well as the Markov property (Section 2.4.3). Similar to Hidden Markov Models, Dynamic Bayesian Networks allow hidden and

observed nodes. The variables in the model may be discrete or continuous, and time itself may be modeled as discrete or continuous as well. However, models with discrete random variables and discrete time are more popular and are computationally more tractable. Dynamic Bayesian Networks can be considered as a generalization of Hidden Markov Models and Kalman Filters. For a detailed description of Dynamic Bayesian Networks, the reader is referred to [21].

This dissertation only considers Dynamic Bayesian Networks with discrete variables and a discrete representation of time. A simplified structure of the Glucose-Insulin Model used as one of the test cases for the research described in this dissertation is shown in Figure 2.7 as a 2-timeslice model. The figure shows a screenshot of the model in Projeny, a tool developed as part of the research described in this dissertation.

This model shows glucose homeostasis modeled as a DBN with complex interactions between various stochastic random variables both within and across timeslices. Insulin resistance, insulin secretion, and total insulin given are hidden nodes, and the rest are observed nodes. All the variables are modeled as discrete nodes.

2.4.7 Partially Observable Markov Decision Processes

A Markov Decision Process (MDP) is a Markov Chain (Section 2.4.1) where some variables denote input or action nodes whose values are provided by a user or an agent. A Markov Decision Process assumes that the entire system is observable to the agent. However, this is hardly the case. For example, the patient's endogenous insulin secretion or insulin resistance are not visible to a closed-loop insulin infusion system or to the clinician who calibrates the insulin dosage. In such cases, the action is based on an observed variable which represents the state of a hidden variable. Such models where the action depends on the observed value of a noisy variable which denotes another observed variable are known as Partially Observable Markov Decision Processes (POMDP). MDPs are also known as Completely Observable Markov Decision Processes (COMDP) to distinguish them from POMDPs.

Automated systems that use POMDPs use a reward variable or function to score past actions and select future actions. In an MDP, the state of the system is updated after each action, since the system is completely observable. In a POMDP, the state of the

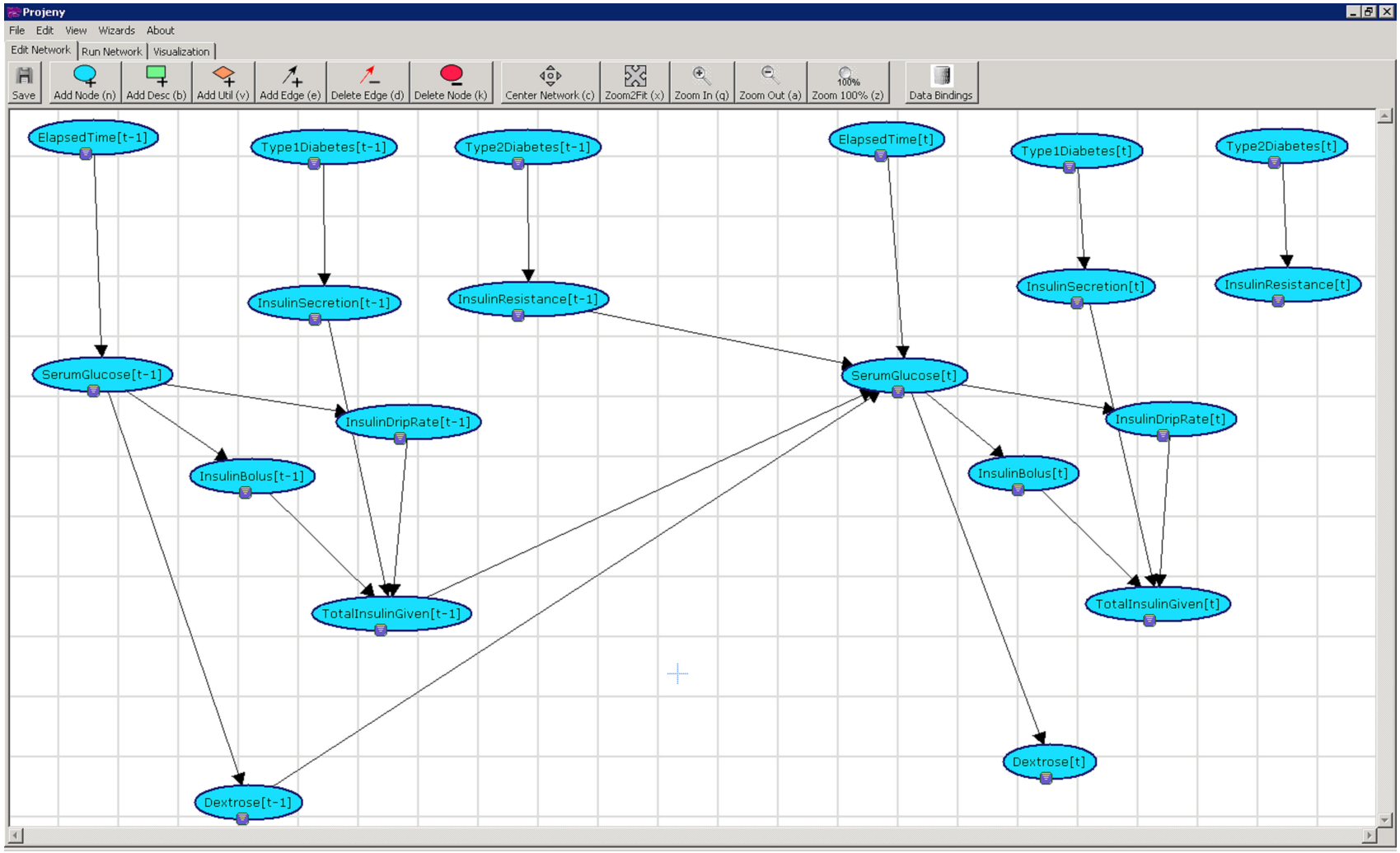


Figure 2.7: Serum glucose and insulin dose DBN model

hidden variable cannot be updated directly after each action, but is expressed in terms of a probability distribution (observation probability). Exact inference for POMDPs is intractable in many cases. However, approximate inference algorithms are available to perform inference with POMDPs. For a detailed description of POMDPs, please see [108, 109, 110].

2.4.8 Limited Memory Influence Diagrams

Partially Observable Markov Decision Processes (POMDP) assume that all the past data about a system are available. However, it is not always possible to obtain all the past information about a system. For example, a diabetic patient may not have the old medical records that provide a historical picture of her Glycosylated Hemoglobin (HbA1c) results. Hence, the model needs to work with limited memory and relax the 'no forgetting' requirement. Lauritzen and Nilsson introduce the concept of Limited Memory Influence Diagrams (LIMID) to model such systems[111, 112]. They also describe methods to find locally optimal solutions and to investigate whether these solutions will be globally optimal[112]. LIMIDs are a recent technique, and applications in the biomedical domain are becoming available[113].

2.5 Inference and Learning with DBNs

Learning is the process by which the structure or the joint probability distributions of the Dynamic Bayesian Network are discovered. After the structure and parameters of a model are known, the model can predict future events or explain past events, in a process known as inference. Learning and inference in case of Dynamic Bayesian Networks use algorithms similar to those used in Hidden Markov Models (HMM), and Kalman Filter Models. Many of these algorithms are temporal extensions of those used for static Bayesian Networks. Learning and inference may be performed online or offline. 'Offline' denotes that all the temporal data corresponding to the model are already available, and learning or inference is done with this fixed batch of data. 'Online' denotes that learning or inference is sequentially updated as new data become available. Restrospective analysis of clinical data are amenable to offline inference. Prospective clinical decision support requires online inference. Learning may be done offline or online in both these cases, or by

initial offline learning followed by periodic online updates. The discussion presented here about learning and inference summarizes a detailed description of learning and inference tasks, algorithms, and their optimizations by Kevin Murphy[21].

2.5.1 Inference

Several inference tasks for Dynamic Bayesian Networks and temporal models in general have been described[21]. All these tasks involve calculating the marginal probabilities of variables of interest. These inference tasks are graphically represented in Figure 2.8, reproduced with permission from Kevin Murphy[114]. In this figure, the shaded region represents the time interval for which data are available. The symbol t represents the current time, T denotes the length of the sequence, X denotes the hidden variable, and Y denotes the observed variable. The hidden variable X emits the observed variable Y described by the observation probability. The upward-pointing arrow denotes the time instance at which we want to perform the inference.

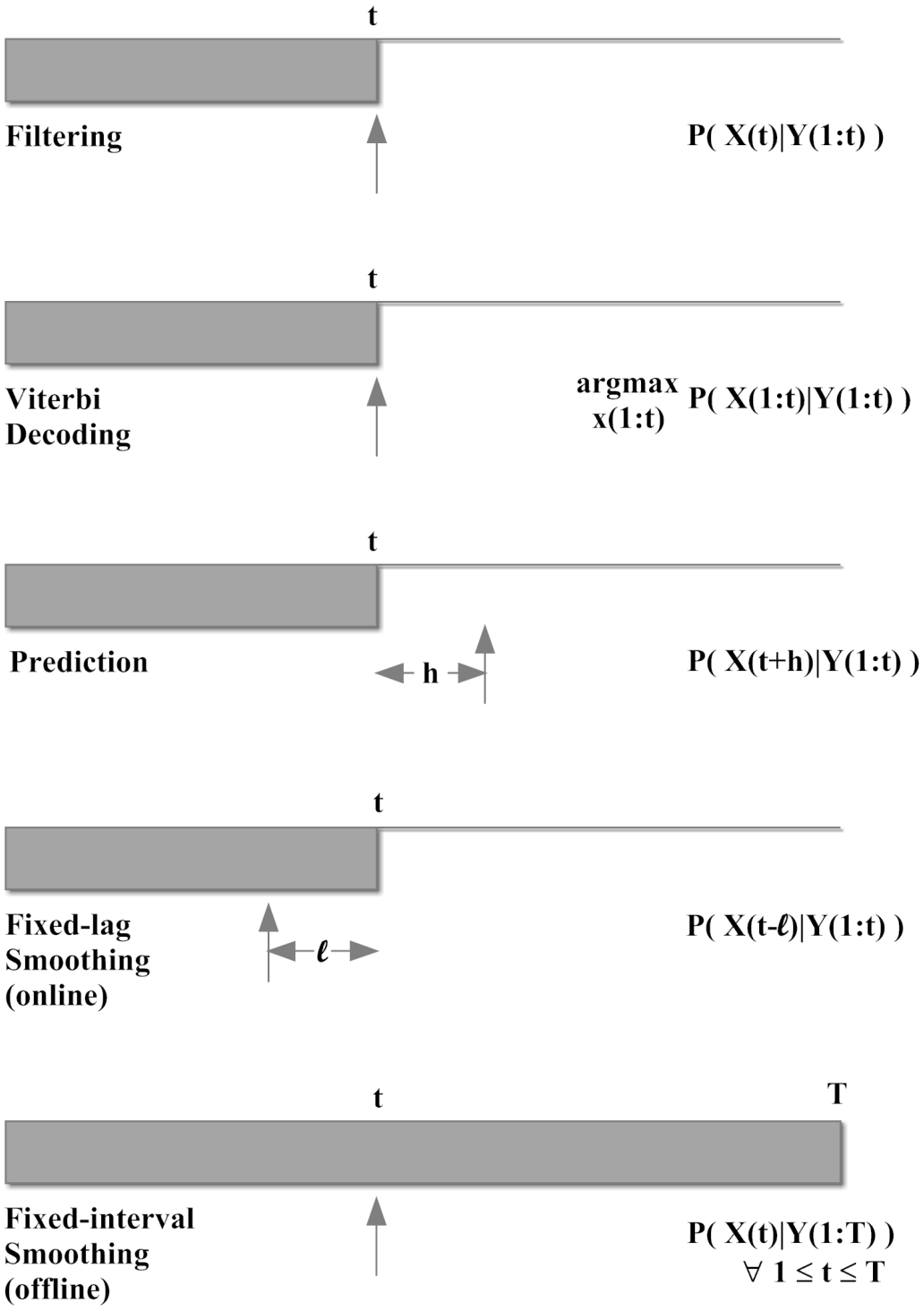
Filtering denotes the estimation of the present state of the hidden variable when the past and present values of the observed variable are known. This process is known as 'filtering' because the observation probability is noisy, and we filter the noise to estimate the hidden variable.

Viterbi decoding denotes the estimation of the most likely sequence of the states of the hidden variable until the present time, if the values of the observed variable up to the present time are known.

Prediction denotes the estimation of the value of the hidden variable at a future point in time, if the values of the observed variable up to the current time are known.

Smoothing denotes the process of estimation of the value of the hidden variable at some point of time in the past, if the value of the observed variable up to the present time is known. For example, we might want to know whether a patient might have had an insulin overdose in the past if we have a record of the historical serum glucose values. Smoothing may be performed online or offline.

Control is performed with Markov Decision Processes (MDP), and Partially Observable Markov Decision Processes (POMDP) where the model includes input or control variables. In these models, an observed variable is set to a desired value, and the value



Legend: Shaded region: time for which data are available. t : current time. T : length of the sequence. X : hidden variable. Y : observed variable. Arrow : time at which inference is performed. Probability term on the right side denotes the inference task to be performed.

Figure 2.8: Inference tasks for temporal probabilistic models

of a sequence of input variables required to produce the desired values of the observed variable are estimated using a reward function. In the case of discrete nodes, control is modeled as an influence diagram. Markov Decision Processes are outside the scope of this dissertation, and are not discussed further. For a detailed description of inference with POMDP, please see [110].

2.5.1.1 Exact Inference

A Dynamic Bayesian Network with only discrete nodes can be converted into a Hidden Markov Model, and the forwards-backwards algorithm can be applied on it for exact inference[105]. The algorithm is efficient as long as the state-space is not very large. For large state-spaces, more efficient methods are required. Frontier algorithm and interface algorithm are two efficient algorithms for this case. The frontier algorithm considers the current timeslice as the frontier which d-separates the past from the future, and performs reasoning[115]. The interface algorithm is an optimization that considers a 1.5 slice DBN - it considers the second slice along with only the temporal nodes from the first slice, and then performs forwards-backwards passes to calculate the marginal probabilities[21]. Similarly, a DBN with linear-Gaussian nodes can be converted into a Kalman filter model and exact inference can be performed with the Kalman Filter model[21].

An alternative approach that applies to all DBN models is the variable elimination technique. The DBN is first unrolled (expanded) for the necessary number of timeslices, and then filtering and smoothing are performed on the unrolled junction tree model[116]. Only two slices of the model need to be stored in the memory at a time to perform junction tree inference.

2.5.1.2 Approximate Inference

Exact inference is slow for models with fully discrete nodes. In addition, exact representations of the state-space do not exist for some continuous, and mixed continuous and discrete models. Approximate inference algorithms are available to perform inference in these cases. Deterministic and stochastic approximate inference algorithms are available.

Deterministic algorithms behave in a predictable fashion, where a given set of inputs always produce the same outputs. For example, the simplest type of a deterministic algo-

rithm is a mathematical function. The popular deterministic algorithms for DBN inference are the Boyen and Koller (BK) algorithm[117], Factored Frontier (FF) algorithm[118], and the Loopy Belief Propagation (LBP) algorithm[119]. Murphy proves that the BK and FF algorithms are special cases of the LBP algorithm[21]. The BK algorithm approximates the joint distribution over an interface as the product of the marginals of smaller terms. However, BK performs exact inference for a two-slice DBN, and becomes intractable for very large state-spaces. The Factored Frontier algorithm provides a more aggressive approximation than the BK algorithm. Approximate algorithms known as the Moment Matching (MM) algorithm[120], and Expectation Propagation (EP) algorithm[121] are available to perform filtering and smoothing in Kalman Filter Models (DBN models which are continuous, linear, and Gaussian). In cases of continuous models that are nonlinear or non-Gaussian or both, the deterministic approximate inference algorithms are still applicable if the posterior can be approximated by a Gaussian. However, for highly multimodal posteriors, stochastic algorithms are more accurate. For both discrete and continuous cases, there is insufficient knowledge about the accuracy of the deterministic approximate algorithms compared to the stochastic approximate algorithms.

Stochastic algorithms do not provide fixed or predictable outputs for a given input. They are based on sampling techniques, and have many advantages over deterministic algorithms. Both offline and online inference algorithms are available. Offline methods include Markov-Chain Monte Carlo (MCMC), with Gibbs sampling and simulated annealing as special cases. Online methods use Particle Filtering (PF), and several variations are available. Stochastic approximate inference algorithms are applicable to a wide variety of models, namely discrete, continuous, or a mixture of the two. Their state-space can have variable size, and the model can change over time. They are guaranteed to give an exact answer with an infinite number of cases. However, these versatile models come with a performance hit. They are unsuitable for very large models or with large amounts of data. However, the speed limitation can be addressed by using a combination of exact and stochastic inference algorithms. This is performed by using exact inference on some of the nodes, and then performing sampling on the rest, known as Rao-Blackwellisation[122]. This can be combined with Particle Filtering, in a technique known as Rao-Blackwellised Particle Filtering. A combination of Rao-Blackwellisation with MCMC methods is also

available[21].

2.5.2 Learning

Bayesian Networks have the desirable property of learning from evidence. Bayesian Networks can learn both the network structure and the conditional probabilities. The former is called structure learning and the latter is called parameter learning. The learning algorithms for Dynamic Bayesian Networks are adaptations of those for static Bayesian Networks.

2.5.2.1 Expectation Maximization Algorithm

Clinical data are often incomplete due to the fact that not all clinical observations are recorded at all time points. Compared to data collection in most other domains such as weather forecasting, financial analysis, genetic or proteomic sequencing, or speech recognition, clinical practice is inherently a data-sparse domain. A clinician may order a lab test on a given day, and may not repeat the test until the history, physical examination, or acuity of illness warrants repeating the test. Even physical examination findings are not performed at every time instance unless necessary. Hence, we need algorithms that can learn the parameters or the structure of the model using sparse data sets.

We begin by describing the Expectation Maximization (EM) algorithm, one of the frequently used algorithms for both parameter and structure learning in both static as well as Dynamic Bayesian Networks that can handle data sets with missing data[123]. The EM algorithm is traditionally used for parameter learning with sparse data. The EM algorithm internally implements a variety of inference algorithms as part of the parameter learning process, and the choice of this inference algorithm can be chosen manually based on the model.

The EM algorithm is an iterative hill climbing algorithm with a two-step process, an expectation (E) step and a maximization (M) step. In the expectation step, the nodes with missing values are filled in based on the values of the observed nodes and the current values of the parameters. In the maximization step, the parameters are recalculated using the filled in values as if they were observed values. The log likelihood of the parameters given the model and the data are calculated after each iteration of the E and the M steps.

This is known as the expected log likelihood, which serves as a surrogate for the log likelihood, since the log likelihood cannot be calculated directly due to missing data. This process is repeated until convergence when the expected log likelihood is maximized.

2.5.2.2 Parameter Learning

Parameter learning in Dynamic Bayesian Networks is similar to that in static Bayesian Networks. Parameter learning may use either Bayesian equations involving conditional probabilities, or a frequentist approach using available evidence. Even if the frequentist methods are used, the model may still be known as a 'Bayesian' model because inference using this model involves Bayesian methods. The parameters of a temporal model are tied across timeslices when the model is assumed to be a homogenous Markov process. This reduces the number of parameters required to describe the model, and permits the model to support training cases with variable or infinite time durations ('lengths') for both learning and inference.

2.5.2.3 Structure Learning

Two types of structure learning algorithms are available, namely constraint-based learning and score-based learning. Constraint-based learning tries to find a model structure that satisfies a set of predefined constraints. Score-based learning uses predetermined scores for specific network substructures to find the structure of the complete model that maximizes the score.

Structure learning involves learning both the interslice and intraslice structures. Intraslice structure learning is similar to that with static Bayesian Networks. The intraslice connections must form a directed acyclic graph. Once intraslice connections are learned, learning interslice connections becomes a variable-selection problem, where the parents of nodes in timeslice t must be chosen for timeslice $t - 1$. For fully observed models with complete data, a variety of structure learning algorithms are available. In cases with missing data and partial observability, structure learning in Dynamic Bayesian Networks becomes intractable.

Structure learning algorithms are available, but are not always tractable. Structure learning uses inference as a subprocess, and a variety of inference algorithms can be

employed which affect both the accuracy and tractability. A structure learning algorithm based on the EM algorithm, known as the Structural EM (SEM) algorithm, has been described[124, 125]. Attempts are being made to apply the structural EM algorithms to learn the structure of temporal models with sparse or missing data[126], though successful implementations are not yet available.

Structure learning need not be a fully automated process. Parts of the structure or the entire structure can be manually defined using domain knowledge, and the best structure can be chosen from these predefined models using the structure learning algorithms. All the models in the experiments described in this dissertation had their structure manually defined using clinical literature.

2.6 Challenges in Temporal Reasoning

The process of temporal reasoning is met with a variety of challenges due to the nature of the temporal reasoning task, the nature of the data and preparing them in an appropriate way, the limitations of the temporal reasoning methods, lack of feature-rich and user-friendly toolkits, and the nature of the medical domain itself. These challenges have impacted the application of temporal reasoning in general, and in the medical domain in particular. The adoption of probabilistic and other temporal reasoning methods in medicine severely lags behind many other domains. We outline the main reasons that impact the adoption of temporal reasoning methods in the medical domain.

2.6.1 Missing Data Problem

As described in Section 2.5.2.1, medical practice is a very data sparse domain. Clinical practice does not require the measurement of all variables at all instances of time. Different data elements are measured at different frequencies and intervals based on the nature of these variables, their past values, and the clinical condition of the patient. Most data are also not measured at constant intervals. The interval between periodic measurements of the same variable fluctuates due to a variety of factors, mostly due to human causes. Physiological parameters such as heart rate and blood pressure can be measured continuously with electronic devices. However, a large number of clinical observations are recorded by human experts and cannot be automated or measured continuously.

Temporal probabilistic models work well if all the data are available at all points in time, and if the time points are evenly spaced. Such evidence cannot be provided by most medical data. The alternative is to develop models, algorithms, and data preparation techniques that support missing data, data being collected at various points in time, and with variable intervals.

2.6.2 Granularity of Time

The frequency of data collection varies depending on the nature of the patient's illness, the patient's clinical condition, and the medical service where the patient is treated. The vital signs (heart rate, respiratory rate, blood pressure) are often measured at 15-minute intervals in the ICU. For critically ill patients, the serum glucose is measured at 2-hour intervals in the ICU. However, for patients who are not critically ill or who are admitted to regular (non-ICU) medical or surgical wards, the vital signs may be collected at 12-hour or 24-hour intervals. Parameters such as serum glucose or other laboratory tests may not be performed every day.

The values of different clinical variables hold true for different durations of time, depending on both the nature of the variable itself and the clinical condition of the patient. For example, serum glucose fluctuates significantly even in normal healthy individuals depending on their food intake. Serum glucose fluctuates to a greater degree in individuals with diabetes mellitus, and in those who are critically ill. Serum glucose measurements do not hold true for more than a few hours, and are hence measured at 2-hour intervals in the ICU, and are often measured once or many times a day by diabetic patients who are not acutely ill. However, HbA1c values do not fluctuate rapidly and provide a moving average of the glucose control status of the patient. This is due to the fact that the average lifespan of the red blood cells (RBC) is about 120-days, and glycosylation of hemoglobin is an irreversible process that requires constant exposure to serum glucose, and is directly proportional to the level of serum glucose over 4 to 12 weeks. Hence, the value of HbA1c will hold true for a longer period of time compared to the value of serum glucose.

Similar variations in measurement of data and validity of the measured data are observed with all clinical data elements. The granularity of time both in terms of measurement of data, and the validity of measured data, introduce complexities in building a

temporal model that explains these data.

2.6.3 Temporal Data Aggregation

From Sections 2.6.1, and 2.6.2, we see that clinical data are measured at variable intervals of time, and each observation is valid for variable durations of time. We also encounter cases where multiple measurements are made within a given interval of time, either because the previous measurement was not available to the clinician who made the second measurement, or the clinician wanted to confirm the previous measurement. In some cases, clinical data are just lost due to human or electronic factors. This leads to issues in cleaning and preprocessing the data to convert it into a suitable format that can be used by a temporal reasoning system. These difficulties pose additional challenges to applying temporal reasoning in medicine. Temporal data aggregation and data preparation are explained in the context of a detailed temporal data preparation framework in Section 3.1.

2.6.4 Challenges with Dynamic Bayesian Networks

In addition to challenges due to the nature of the medical domain, and the nature of the clinical data themselves, there are additional challenges posed due to the requirements and limitations of the temporal reasoning methods and techniques. Challenges posed due to the constraints of Dynamic Bayesian Network methods and techniques are described below.

2.6.4.1 Higher Order Markov Processes

Many Dynamic Bayesian Network toolkits assume that the system being designed involves first-order Markov processes. This is rarely the case with clinical processes. Different clinical processes in a system may have different temporal orders. For example, serum glucose may only depend on the feeding and insulin dosage data over a short period of time in the past. However, serum HbA1c depends on serum glucose for a long period of time in the past. It is hard to estimate current HbA1c value using a HbA1c value measured 3 months ago, and just the latest serum glucose value.

2.6.4.2 Structure Learning Problems

Dynamic Bayesian Networks require a causal or explanatory structure for the DAG model. The graph structure may be described based on medical literature, or discovered from available data, or by a combination of the two. Description of biological processes in medical literature are not easy to capture in a Dynamic Bayesian Network. The hidden nodes in a model are not easily understood from the medical literature. One would need to combine literature from medicine, pathology, physiology, biochemistry, pharmacology, and so on before the structure of the model along with its hidden nodes becomes apparent. This is a very imprecise process. Clinical practice guidelines are often not suitable for modeling in a probabilistic network. This claim is demonstrated in Chapter 5 while comparing different models created by the author to predict sepsis in the emergency department.

Temporal structure learning algorithms have various limitations described in Section 2.5.2. These limitations, combined with the imprecise nature of medical knowledge, make structure learning a significant challenge to temporal reasoning in medicine.

2.7 Relevant Works Involving DBNs in Medicine

We present a comprehensive overview of works involving temporal reasoning in medicine that are relevant to the methods and experiments described in this dissertation. In spite of the various challenges involving temporal reasoning in medicine, many works involving temporal analysis and time-series prediction have been published. These works use a variety of techniques, from rule-based, regression-based, and neural network-based techniques on one end, to probabilistic techniques such as Markov Models, static Bayesian Networks, and Dynamic Bayesian Networks on the other end. They involve problems on a short timescale such as prediction of the serum glucose, to a long timescale such as prediction of cancer survival and transplant graft survival. Some experiments involve simulated data, whereas some experiments use real clinical data. Experiments involving simulated data can be used to validate the methods, whereas experiments involving real data serve as validation and proofs of concepts of these methods' ability to answer clinical problems. In this section, we only include works involving temporal modeling and prediction in the biomedical domain. Temporal representation, temporal databases, and temporal query

languages are not discussed.

Dynamic Bayesian Networks have been applied in medicine in a very small number of cases. Some of these studies have used simulated data, and some have used very limited amounts of real clinical data. Very few studies have used large data sets comprised of real patient data. However, other temporal probabilistic modeling techniques such as Hidden Markov Models have been used extensively in medicine. Other modeling techniques such as artificial neural networks and logistic regression models have also been used in medicine extensively. In December 2009, a PubMed query for 'Hidden Markov Models' returned 1,443 results, whereas 'Dynamic Bayesian Networks' returned 46 results. Less than 10 of these 46 articles were relevant to clinical medicine, and less than 5 involved large patient data sets. Atemporal models using Bayesian Networks have also been used in medicine to a large extent, with some using special nodes in a static model to represent temporal data.

One of the earliest works describing the use of Dynamic Bayesian Networks in the biomedical domain is by Andreassen et al. in 1991[15]. Andreassen et al. described a combination of Dynamic Bayesian Networks and differential equations to model serum glucose and insulin dosing, and applied it to a data set consisting of 12 patients with insulin-dependent diabetes mellitus using Hugin, a proprietary Bayesian modeling toolkit. Dagum and Galper described a Dynamic Bayesian Network model to predict sleep apnea in a single patient using a large data set in 1992[127]. Hernando et al. described DIABNET, a temporal causal probabilistic network model used to make qualitative recommendations on insulin therapy for patients with gestational diabetes, in 1996[128]. Hernando et al. have since described an evaluation of the DIABNET system using the data of 9 patients[129].

Leong has described a temporal probabilistic modeling language named DynaMol and theoretical models of various clinical scenarios[130]. Provan and Clarke described DYNASTY, a system that enables construction of dynamic temporal probabilistic models using some clinical examples[131].

Galan et al. described NasoNet, a temporal probabilistic system for modeling the spread of nasopharyngeal cancer[132]. Sebastiani et al. have described a study about using Dynamic Bayesian Networks to detect influenza in a pediatric emergency department[133]. Xiang et al. describe miniTUBA, a web-based system that uses Dynamic Bayesian Net-

work models for clinical decision support[134]. Dynamic Bayesian Networks have been used by van Gerven et al. to predict the prognosis of patients with carcinoid tumors[135]. Charitos et al. have described a Dynamic Bayesian Network system to predict ventilator associated pneumonia using a data set of 20 patients[136]. Langmead has described the use of Dynamic Bayesian Networks to make treatment decisions using simulated data of patients with sepsis[137]. Peelen et al. have described a Dynamic Bayesian Network model for predicting the outcome of patients with sepsis in an intensive care unit[138]. This study used Dynamic Bayesian Networks with a large data set (2,271 patients), demonstrating successful use of DBN in a clinical setting.

In this dissertation, we describe the theoretical foundations, challenges, methods to overcome the challenges, and a toolkit to enable the application of DBN in medicine. We hope that this detailed discussion will help to improve the use of Dynamic Bayesian Networks in medicine.

CHAPTER 3

MATERIALS AND METHODS

The experiments described in this dissertation involve several steps to perform temporal modeling and prediction, starting with clinical data from the electronic record. The temporal reasoning methods described in this dissertation are applied to two test cases - insulin dosing and glucose homeostasis in the ICU, and early prediction of sepsis in the emergency department. Multiple models were built and experiments were performed for each of these two test-cases. This chapter describes the general materials and methods that are common to both test-cases and their respective models and experiments. The specific materials, methods, and results of each model and experiment pertaining to the two test-cases are described in further detail in Chapters 4 (insulin-glucose test-case), and 5 (sepsis test-case).

The data first need to be transformed and abstracted into a format suitable for temporal reasoning, while minimizing the loss of information due to these transformations. Various temporal models need to be constructed that reflect the disease and decision processes being modeled. The tools and algorithms involved in the experiments are described in brief. The computational complexity and predictive accuracy of various models and data preparation methods are briefly described, with further details in Chapters 4 and 5.

3.1 Data Preparation

The accuracy of predictions performed by a machine learning system depends on the quality of data provided to the system. The input data determine the accuracy of the model and the technique, because many machine learning systems learn the structure of the model, the parameters, or both from the training data. As shown in Chapters 4 and 5, the quality of training data can determine the success or failure of a model or a technique. Machine learning systems such as Dynamic Bayesian Networks cannot directly

use the raw data obtained from an electronic medical record system. The data need to be preprocessed and transformed into an appropriate format before they can be used by a Dynamic Bayesian Network-based model or system.

Probabilistic machine learning models require data in a continuous or discrete format. They cannot use unstructured or free-text data. The models described in this dissertation require discrete data. Clinical data from various sources need to be compiled together and transformed into a time-stamped, discretized format with a common structure for use by automated tools. In this section, we describe the nature of the source data, and the transformations and preprocessing that need to be performed before the data are usable by our Dynamic Bayesian Network models.

3.1.1 Data Aggregation and Abstraction

Clinical data are generated by a variety of sources in a variety of formats. Clinician (physician or nurse) charting provides the history and clinical examination data in a partly structured and partly free-text format. Patients' history is often free-text, although some electronic medical record systems encode these data using structured data entry in the form of type-ahead prompts or pick lists. Vital signs are generally captured using a structured data entry form and encoded using biomedical terminologies.

Laboratory data are usually encoded using a clinical terminology, which may be a standard terminology such as Logical Observation Identifier Names and Codes (LOINC), or a local terminology developed by the specific laboratory. Data generated by automated monitoring devices are often numeric data wrapped in a standard messaging syntax such as Health Level Seven (HL7) messages. Medication orders are captured using structured data entry forms using standard terminologies as well. The medication order and administration records often use a fine-grained information model, which allows the drug's brand name, physical form, route of administration, ingredients, strengths of various ingredients, the dosage and frequency of administration, and the time of order and administration to be queried from the clinical database. A data mining method that takes advantage of the information models used to capture, encode using clinical terminologies, and store these structured data in the database is highly desirable, but is not currently available. A tookit with this capability would allow the researcher to extract composite clinical information

using the same structure with which it was recorded by the clinician.

Structured and encoded electronic data are not available in all hospitals. Even in hospitals with advanced electronic medical record systems, a large portion of data are available only on paper records. These records may be scanned and stored as images in the electronic medical record system, which do not support automated querying and data retrieval.

At Intermountain Healthcare's LDS Hospital in Salt Lake City, the data are captured and stored in a well-structured form using an information model and an enterprise reference terminology. The original electronic medical record system, HELP (Health Evaluation through Logical Processing) encodes the data using a hierarchical data dictionary known as PTXT (Pointer to text), and stores the data in the HELP database in mostly encoded and partly free-text forms[139]. Intermountain Healthcare also has a newer electronic medical record system known as HELP2, which has a multihierarchical concept-based Healthcare Data Dictionary (HDD). The HELP2 data are stored in a Clinical Data Repository (CDR). Both the HDD and CDR were developed in collaboration with 3M Health Information Systems[140].

The data from HELP and HELP2 are highly structured and encoded using biomedical terminologies, and are usable for clinical documentation as well as decision support. However, all the data required for the temporal probabilistic models needed to be aggregated and abstracted before they could be used by the temporal reasoning tools. The procedures described in this section apply to both the insulin-glucose and the sepsis prediction models. The differences in data preparation between these two test-cases are noted where appropriate.

3.1.1.1 Data Aggregation

The data required for both the insulin-glucose models and the sepsis prediction models were available in encoded form in the clinical data repository. However, different clinical variables required for these models were in disparate database tables, and were encoded in different formats. For example, the patient's age, date of admission, and admit diagnosis were in the clinical encounters table. The laboratory results were all available in a single laboratory results table, and the vital signs were all available in another table. Medication

administration data were in a separate table. To train a predictive probabilistic model, medication administration was more appropriate than the medication order data, since the medication orders were not always in agreement with the administered medications.

The raw clinical data were available in an entity-time-attribute-value table format. Each row in the laboratory or vital signs table had columns identifying the patient and the date/time-stamp of the observation. The tables had additional columns that specified the data element and the value of the data element. The name, attributes, or the value of the observation may be contained in a single column each, or spread across a group of columns. If the observation is a simple data element such as serum glucose, it may be contained in a single column. For cases where the data element conveys complex information such as the serum glucose measured 1 hour after administration of a glucose oral dose, performed as part of a glucose tolerance test, this information would need to be postcoordinated from multiple pieces of observation. Similarly, simple data elements such as binary or nominal values were stored in a single column. However, numeric data elements were spread between multiple columns, with the value in one column and the units of measure in another. An analysis of the PTXT terminology and the Healthcare Data Dictionary were performed and compared with the data stored in the clinical data repository to combine multiple pieces of clinical data from different rows or columns in the database table to reconstitute meaningful pieces of clinical information.

A denormalized table format was required to support the temporal reasoning tools used in our experiments. The denormalized table had two columns representing the patient identifier and the date/time identifier, respectively. However, the remaining columns were not in the attribute-value format as in the source data tables. The denormalized table had multiple additional columns, each representing a single, meaningful reconstituted clinical variable. For example, in the case of the insulin-glucose models, in addition to the patient identifier and date/time identifier columns, there were additional columns representing the serum glucose, current insulin IV drip rate, current insulin IV bolus dose, current dextrose dosage, patient's diabetes status, etc.

Different clinical variables were measured with different frequency and periodicity in the clinical setting. For example, for critically ill patients in the ICU, vital signs were measured once every 15 minutes to an hour, and the serum glucose was measured once

every 2 hours. Lab tests were performed less often. Data elements that were measured together did not have the same date/time-stamp in some cases. They were often 1 to 15 minutes apart. Hence, storing them in the denormalized table produced several rows where only a handful of columns were populated. For each patient, we first loaded the timestamp and the values of the most numerous clinical variable into the denormalized table. We then selected the second most numerous clinical variable. If the patient identifier and timestamp of a given row of this second clinical variable existed in the denormalized table, we updated the row in the denormalized table to store the value of this second clinical variable in its own column. If the combination of the patient identifier and the timestamp did not exist, then a new row was inserted into the denormalized table with this value. This process was repeated for all clinical variables in the data set.

3.1.1.2 Temporal Abstraction

At the end of the data aggregation step, all the data reflecting the clinical variables in the model are stored in a single denormalized data table. The data present in this table are used to train and test the model. However, the data rows differ by a few minutes to a few hours, and produce a very sparse data table. A very sparse data table when used for training necessitates the use of the expectation maximization (EM) algorithm which increases the computational expense of the model while reducing the accuracy of the learned parameters. However, the data can be temporally consolidated to pick one representative data point per time interval for the smallest time interval represented in the model, which will reduce the need for imputing missing values using the EM algorithm.

The smallest time interval to be supported by the model is based on both the nature of the model and the availability of data. In the case of the insulin-glucose data set, the serum glucose is measured and the insulin dose is adjusted once during every 2-hour interval for critically ill patients in the ICU under the current insulin dosing protocol (eProtocol-insulin) in use at LDS Hospital. Hence, a good starting point was to abstract the data to select one representative measurement for each clinical variable in the model for every 2 hours to coincide with serum glucose and the insulin drip rate measurements. In the case of the sepsis data set, which consisted of both cases and controls from the emergency department, the model consisted mostly of vital signs, which were available once every

hour in most cases. So, a timeslice interval of 1 hour was chosen for the sepsis models, and a representative data point was chosen for each clinical variable during every 1-hour interval.

In both the insulin-glucose and sepsis prediction data sets, we encountered both multiple instances and no instances of various clinical variables observed during each chosen timeslice interval. Missing data can mean a variety of things: the data were not measured, measured and then lost, or they were uneventful and in line with the expected values given the prior measurements, and hence not recorded in this case. It may also mean that a value was measured on paper or was stored in a different part of the electronic medical record system, and hence unavailable at the time of data preparation. Several approaches have been discussed to define and overcome the missing data problem. Little and Rubin classify reasons for missing data as missing but completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR)[141]. Lin discusses methods for dealing with missing data which are applicable to the clinical domain, including creating a discrete state for the clinical variable to represent missing data, or creating a separate proxy variable for each clinical variable to represent missing data[142][143].

We did not apply special treatment for missing data in our experiments. Bayes Net Toolbox (BNT), which implemented all the algorithms we needed to train and test the models, supported parameter learning with missing values using the Expectation Maximization algorithm. Hence, we were able to leave missing values as null values in the database, and we designed our temporal modeling toolkit, Projeny, to support null values from the database and to call the expectation maximization algorithms in BNT.

However, we had to choose a representative data point if a clinical variable had multiple observations in a given timeslice interval. A clinician or a laboratory may measure or record multiple observations for a clinical variable due to a variety of reasons. Common reasons for clinicians to make multiple measurements include cases where the initial measurement is not available, either due to a temporary availability issue at the point and time of care, or because the initial measurement was permanently lost. Permanent loss of the data also leads to the missing data problem; however, this cause is often indistinguishable due to the sparse nature of clinical data. Another common reason for a duplicate measurement is to verify and confirm a suspicious initial measurement. The

cause of the duplicate measurement cannot be inferred with confidence unless the reason is recorded by the person or the system that recorded the duplicate measurement, which is typically not done. Hence, we decided to treat all the duplicate data with equal validity, and apply automated techniques to select a representative data point.

Approaches to choosing the representative data point for a variable if multiple data points are available in a temporal data set are discussed by various authors under the context of temporal data clustering[144][145][146]. Some simple approaches for temporal data sampling include selecting the average, or selecting the most abnormal measurement. We decided to select the average, since this process can be automatically applied for all numerical variables. We did not encounter multiple measurements for non-numeric values in our data set.

At the end of this process, we had a temporally abstracted denormalized table, with one data point or a null value for each variable per timeslice per patient. The numerical data were continuous in this data set. The data can be used in this form with models and algorithms that support continuous data. However, we designed our models with entirely discrete nodes, since models with discrete nodes are computationally less expensive and more tractable than models with continuous data. Hence, we chose to discretize the continuous variables in our data set.

3.1.2 Data Discretization

Several data discretization methods are available to discretize continuous data for use with machine learning algorithms. We discuss some of the simple and more popular ones, as well as the more complex but less popular ones. We also discuss how different discretization techniques affected the accuracy and tractability of our models. A detailed review of various discretization techniques is presented in [147], and [148].

3.1.2.1 Data Clustering and Visualization

It is helpful to evaluate the distribution of the continuous variables before we choose the discretization technique and any manually selected cut-off points. We used histograms and cluster analysis to find clusters and study the distribution of each continuous variable individually. Visually discernible multiple clustering was not found for the continuous

variables, and many continuous variables formed one large cluster each with few outliers.

We describe various discretization algorithms that were used in our experiments. Equal frequency discretization was not used in our experiments, but is discussed here for comprehensiveness and its appropriateness. Discretization was performed before the data were divided into training and testing data sets.

3.1.2.2 Equal Interval Discretization

Equal width or equal interval is the most basic method among various discretization methods. The range of the numerical values of the variable of interest is divided into the desired number of intervals or bins by dividing the range equally. The width of each interval, in other words, the difference between the lower and the upper bounds, was the same for all the bins. We used equal interval discretization in combination with domain-based discretization for one of the models for the insulin-glucose test case, as described below in Section 3.1.2.4. Outliers and extreme values were included into the first or last interval, such as '20 and above'. The impact of including the outliers into the first and last bins was not studied. We present the intervals chosen for insulin drip rate using equal interval discretization in Table 3.1.

Table 3.1: Equal interval discretization for insulin drip rate

State number	Insulin drip rate in U/hr (left-open right-closed intervals)
1	0 - 2
2	2 - 4
3	4 - 6
4	6 - 8
5	8 - 10
6	10 - 12
7	12 - 14
8	14 - 16
9	16 - 18
10	18 - 20
11	20 and above

3.1.2.3 Equal Frequency Discretization

A popular alternative to equal interval discretization is equal frequency discretization. In this method, the observed range of the numerical variable is divided into a desired number of bins such that all bins have the same number of observations. The widths of different bins vary, in contrast with equal frequency discretization. The cut-off points for the intervals in equal frequency discretization change based on the data set. This will lead to intervals that are sensitive to each data set, and to noise in the data. The model will require retraining and validation for every new data set, due to changing intervals. Hence, we did not use equal frequency discretization in our experiments.

Hulst compares the error rate between two experiments performed using a Dynamic Bayesian Network model trained and tested using equal interval and equal frequency discretization techniques[149]. The model involved glucose homeostasis and insulin dosing using simulated data of 1,000 patients with no missing data generated using a noisy simulation algorithm. The tests demonstrated that equal interval discretization produced a smaller error rate than equal frequency distribution, and hence, equal interval discretization technique was more accurate in this experiment.

However, it must be noted that Hulst's experiment used very small timeslice size (15 minutes) compared to our insulin-glucose models (2 hours), and Hulst's experiments had no missing data, in addition to using simulated data instead of real clinical data. Hence, the applicability of Hulst's findings to our models cannot be ascertained.

3.1.2.4 Domain-based Discretization

We propose a novel discretization technique named 'domain-based discretization' for discretizing clinical variables. The technique is called 'domain-based discretization' because the discretization is based on clinical domain knowledge about the specific variable. Most physiological parameters have their normal, high, and low ranges defined in clinical literature. For example, serum glucose has very low, low, normal, high normal, high, and very high values described as numerical ranges in clinical literature. We used the normal, high, low, very high, and very low range boundaries defined in clinical literature to discretize various clinical variables for which such a definition is available. We modified the ranges slightly to support the available range of data. The width of the bins became

progressively wider as one moved further away from the normal range of a variable. For example, for discretizing serum glucose, we used for the intervals shown in Table 3.2 by applying domain-based discretization.

As shown in Table 3.2, the width of the intervals was small in the normal range, and became progressively wider as the serum glucose value moved further away from the normal range. The rationale behind this sliding scale is that the clinician would care about small differences within and near the normal range, but would not care about differences of the same small magnitudes as the clinical parameter became extremely high or extremely low.

Domain-based discretization was performed for serum glucose for one of our insulin-glucose models. For parameters that did not have clearly defined high, low, or normal ranges in medical literature, such as the insulin drip rate, we used equal interval discretization for the same model. Thus, a combination of domain-based and equal interval

Table 3.2: Domain-based discretization intervals for serum glucose

State number	Serum glucose in mg/dl	Bin Width	High / low / normal
1	0 - 20	20	.
2	20 -35	15	Very low
3	35 - 50	15	.
4	50 - 65	15	low
5	65 - 80	15	.
6	80 - 90	10	Normal
7	90 - 100	10	Normal
8	100 - 110	10	High normal
9	110 - 120	10	.
10	120 - 135	15	.
11	135 - 150	15	High
12	150 - 170	20	.
13	170 - 200	30	.
14	200 - 240	40	.
15	240 - 290	50	.
16	290 - 350	60	Very high
17	350 - 420	70	.
18	420 - 500	80	.
19	500 - 600	100	Extremely high
20	600 and above	undefined	.

discretization was used for different variables in one of the experiments with the insulin-glucose model. This is referred to again in context of the experiment in Chapter 4.

Training and testing the model with a combination of domain-based and equal interval discretization took a significant amount of time and computer memory, as described in Chapter 4. We found that this combination of equal interval and domain-based discretization produced results with poor accuracy, as shown in Chapter 4. Hence, we applied an information content-based technique known as k-means clustering, described below.

3.1.2.5 K-means Clustering

K-means clustering is an algorithm invented by Lloyd for pulse-code modulation (PCM) in signal transmission in 1957 and published in 1982[150]. Pulse-code modulation is a technique where an analog signal's amplitude is sampled at uniform intervals, and encoded for transmission using a finite set of digital signals. The purpose of the algorithm is to divide a given number of observations into k clusters, where each observation belongs to the cluster whose mean is nearest to the given observation. The algorithm serves to fit n symbols into a channel of width k with minimal loss of information. The technique can be used to cluster both scalar, continuous observations as well as d-dimensional real vectors composed of multiple observations. If n number of d-dimensional real vectors need to be classified into k clusters, the problem can be exactly solved in $O(n^{dk+1} log\, n)$ time.

We used the k-means clustering algorithm implemented in Weka[151]. The algorithm takes k, the desired number of clusters, as the input, and divides the input data into k-clusters such that each observation is in the cluster whose mean it is closest to. The algorithm returns the boundaries of each cluster and the error rate as output. We arbitrarily performed k-means clustering for variables in the insulin-glucose model with $k = 10$, and $k = 15$ clusters. If there was a significant decrease in the error rate using $k = 15$ compared to $k = 10$, we used 15 clusters for the variable. Otherwise, we used 10 clusters.

We selected a smaller number of states with k-means clustering compared to the combination of domain-based and equal interval discretization. This reduced the size of the state-space, and hence the number of parameters and the computational complexity of parameter learning. We found that the insulin-glucose models produced much higher accuracy with the data set discretized by k-means clustering compared to the combination of

domain-based and equal interval discretization. The glucose control model using k-means clustering where the model recommended the insulin dosage performed as well as the current rule-based protocol, as explained in Chapter 4.

We also used k-means clustering for the first sepsis prediction model, as shown in Chapter 5. We found that the model was computationally expensive, and the area under the ROC curve attained by the model was around 0.7. Hence, we explored other discretization techniques and we applied the Minimum Description Length (MDL) model.

3.1.2.6 Minimum Description Length Discretization

K-means clustering only considers the input data set and number of clusters k used to represent the input data. An inappropriate number of clusters will yield poor clustering and lead to loss of valuable information contained in the input data. The optimal number of clusters is hard to determine without experimentation with various values of k for the given data set. Hence, we explored algorithms that preserve information content. We found that the minimum description length (MDL) algorithm described by Fayyad and Irani (1993) finds the minimum number of clusters of the input variable required to describe the variation in the output variable[152]. All the relationships in our models were directed, hence, describing the variation in one variable using variation in one or more variables is straightforward. The variations in the children nodes can be explained in terms of variation in parent nodes. Hence, the MDL algorithm seemed to be appropriate for discretizing the variables in our models. The model works by sorting and then cutting the distribution of input values at specific cut-points that reduce the class entropy of the resulting classes[152].

We found that MDL discretization produced a much smaller number of discrete states for most of the continuous variables in the sepsis model. The glucose homeostatis experiments were conducted a year before the sepsis experiments, and the MDL algorithm was not used at that time. The sepsis model trained and tested using MDL discretization also completed training and testing in a smaller amount of time, and produced higher accuracy than the model that used k-means clustering. However, the sepsis model structure was also changed between the two experiments, and hence the improvement due only to the MDL algorithm was not determined.

3.2 Creating, Training, and Testing the DBN Model

Creating a Dynamic Bayesian Model consisted of three steps: defining the nodes, defining the edges, and defining the states of the nodes. We created a tool named Projeny, written in Java programming language, that allows the user to create, train, and test a model through a graphical authoring interface. Projeny is based on an open-source tool named Bayesian Network Tools in Java (BNJ) by Hsu et al.[44], which appears to have ceased development in 2004. Projeny allows the creation of nodes and edges, along with the states of the nodes. Bayes Net Toolbox (BNT), an open-source Matlab toolbox by Murphy[153], is used to perform the learning and inference tasks for our models. BNT runs inside Matlab, and requires the model and the data to be input and output using the Matlab scripting language. However, Matlab cannot be called from a Java application directly. Hence, we used another open source library known as JMatLink created by Müller[43] to call the Matlab engine from the Projeny Java application. Projeny was developed as part of this dissertation research, and has been released under the GPL (GNU General Public License) version 2 open-source license[154]. Matlab is a proprietary mathematical toolkit published by Mathworks[155]. A user guide of Projeny is provided on the Projeny website[154]. This section uses the insulin-glucose model illustrated in Figure 3.1 to describe the methods. The specific models used in various experiments are described in detail in Chapters 4 and 5.

BNT supports only Bayesian Networks with first-order Markov processes modeled as a two-timeslice Dynamic Bayesian Network. To comply with this limitation, the model is created as a two-timeslice model in Projeny. The structure of the models used in all the experiments described in this dissertation are based on domain knowledge gathered from medical literature. Structure learning algorithms are not used. We first begin by modeling the nodes in the DBN. We then define the states based on the output from the discretization algorithm. Finally, the edges are defined. Intraslice edges are defined first, followed by interslice edges. Projeny generates the conditional probability tables for all the nodes based on their parents, and initializes them with zeroes. For nodes without any parents, probability tables reflecting their prior probabilities are created and filled with zeroes.

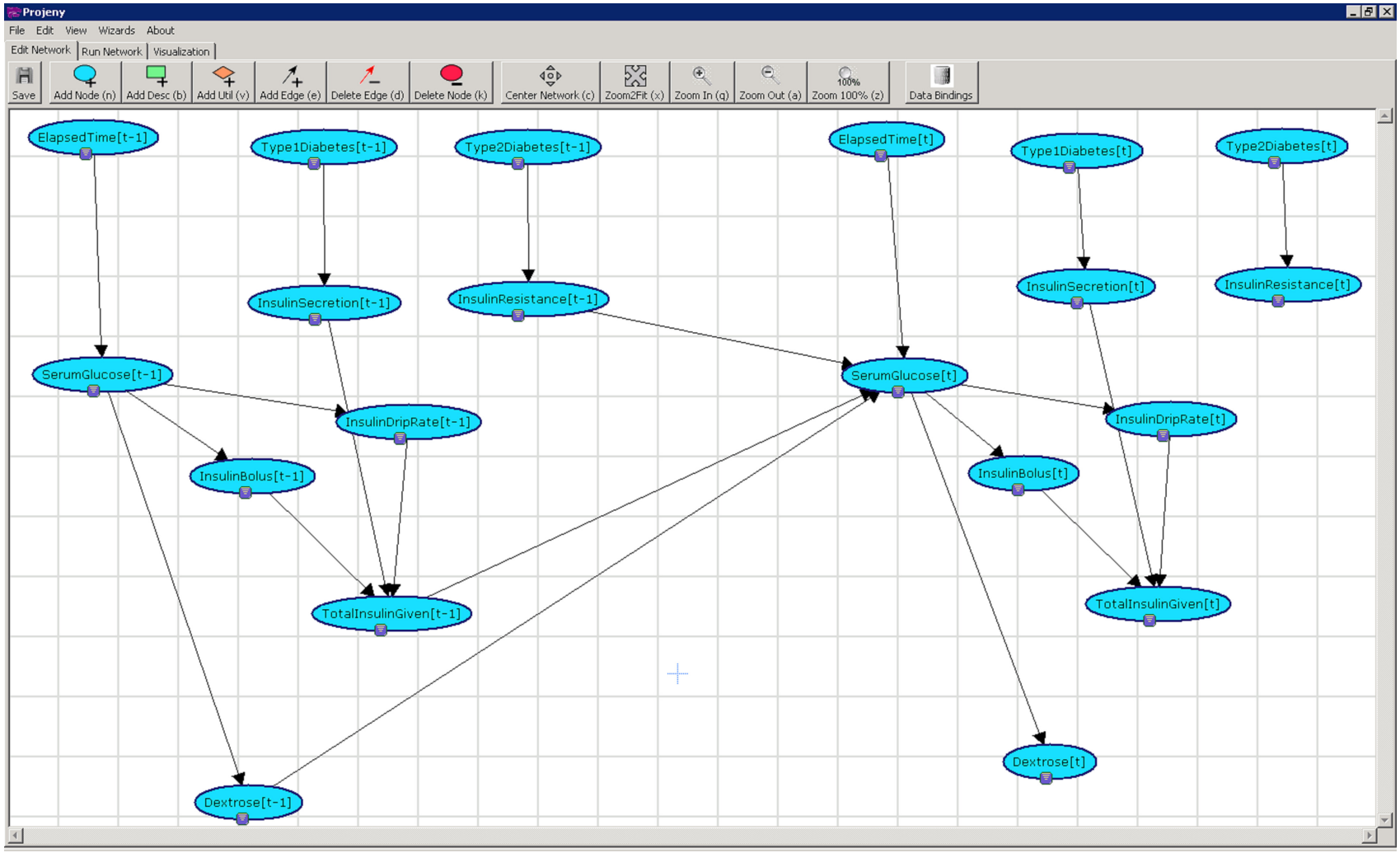


Figure 3.1: Serum glucose and insulin dose DBN model

3.2.1 Training and Testing Data Sets

Separate training and testing data sets needed to be created. The training data set was used for parameter learning, and the testing data set was used for inference. A common convention is to use two-thirds of the data set for training, and one-third of the data set for testing[156]. We took the discretized data set described in Section 3.1.2, and selected one-third of the anonymized patient identifiers ('FakeMRN') at random. The data of these patients constituted the testing data set. The data of the remaining two-thirds of the patients constituted the training data set. The same procedure was performed for all the experiments for both the glucose homeostasis and sepsis models.

Cross-validation is a technique which involves repeated sampling of the full data set, with or without replacing previously derived data subsets into the full data set. Different training and testing data sets are derived during each sampling iteration. The results calculated with different samples are then compared. Cross-validation helps to overcome biases due to sampling errors, and is also very useful with small data sets[156]. All our experiments had large sample sizes, as described in Chapters 4 and 5. Hence, cross-validation was not performed.

3.2.2 Equivalence Class and Parameter Tying

An equivalence class denotes the nodes which have the same set of parameters - conditional probability tables in the case of our models. The parameters in this case are said to be 'tied'. Parameter tying is a benefit of the Markov property by assuming the model to be a homogeneous Markov chain, i.e., the conditional probabilities do not change over time. If a given node in the first timeslice and its cohort in the second timeslice have the same set of ancestors, then they are considered to be in the same equivalence class; if not, they are then in different equivalence classes. Hence, for a model with n nodes per timeslice, the maximum number of parameters (conditional probability tables) to be learned is $2n$, which is the number of nodes in the two timeslices. However, if m nodes in the first timeslice are in the same equivalence class as their cohorts in the second timeslice, then the total parameters needed to describe the model is $2n - m$, since these nodes in the first and the second timeslices have the same parameters. Figure 3.2 shows an insulin-glucose model with the equivalence classes identified. Different

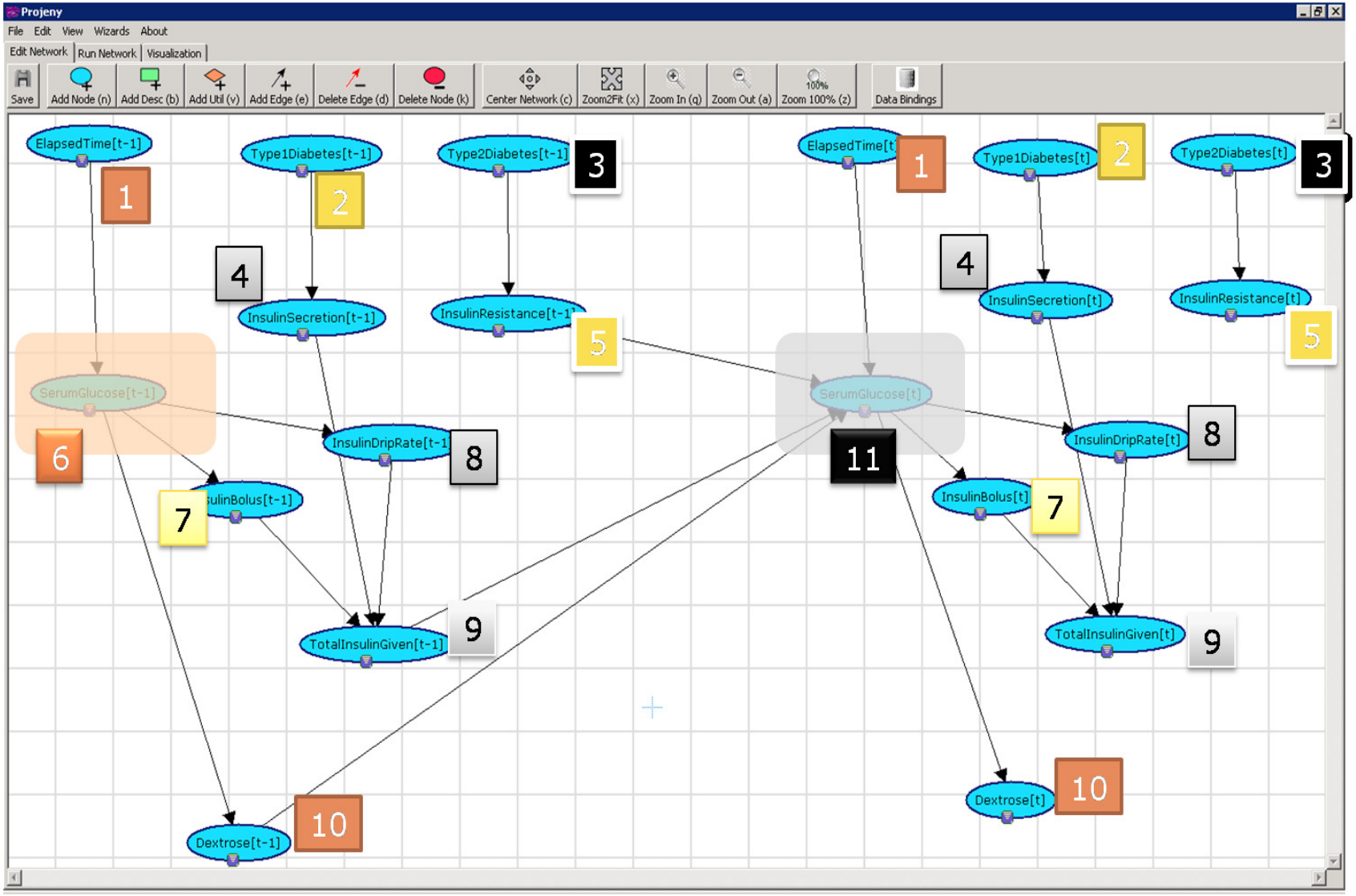


Figure 3.2: Equivalence classes and parameter tying

equivalence classes are shown using numbers with different color schemes next to the nodes. Nine nodes in the first timeslice and nine nodes in the second timeslice are in the same equivalence classes. As shown in the figure, this model has only $2n - m = 11$ equivalence classes.

During training and testing, Projeny automatically unrolls (repeats the network structure) the DBN for as many timeslices as are in the current patient's data. The structure of the nodes and edges are repeated, and the parameters (conditional probability tables) are shared from the second until the last timeslice. Hence, for a model with t timeslices and n nodes per timeslice, instead of having nt parameters, we have $2n - m$ parameters to learn. This reduces the number of parameters to be learned by a factor of t, which reduces the computational complexity for learning. For example, if the model in Figure 3.2 were to be trained on data sets with 10 timeslices, without parameter tying, 100 parameters must be learned. With parameter tying, the model above has only 11 equivalence classes. Hence, only 11 parameters need to be learned whether the data set has 10 or more timeslices.

In addition to the reduction in complexity, parameter tying also helps to support training and testing data with an arbitrary number of timeslices, and a data set where each case (patient) has data with a different number of timeslices. This provides support for real clinical data where different patients have different lengths of stay, and hence data sets of different temporal lengths. An alternative to this approach was described by Cho and Haug using a sliding window model to tie the parameters implicitly, and to support models with an arbitrary number of timeslices[157]. However, our approach provides a more straightforward and explicit way to tie the parameters. Projeny identifies equivalence classes and ties parameters automatically, and hence eliminates additional work on the user's part every time the model is modified.

3.2.3 Hidden and Observed Nodes

In the Hidden Markov Model paradigm, hidden nodes are those that cannot be observed directly by an observer, and are estimated indirectly through another proxy variable or a combination of variables. In the DBN model, a hidden node may be estimated by multiple observed nodes to which it is causally related. For example, in the glucose-insulin model described in Figure 3.1, insulin secretion and insulin resistance are both hidden

nodes. These two nodes describe physiological phenomenon in the insulin and glucose metabolism and they cannot be measured directly. However, they can be estimated from the observed nodes to which they are causally related, including the patient's diabetes mellitus status, the serum glucose, and the insulin dose of the patient. They are easily declared as hidden nodes through the data binding dialog in Projeny, shown in Figure 3.3.

When the data binding dialog box is opened, Projeny lists the names of all the nodes in the first timeslice, and allows the user to select the database column that provides the data for this node. The user begins by connecting to the database. The user first selects

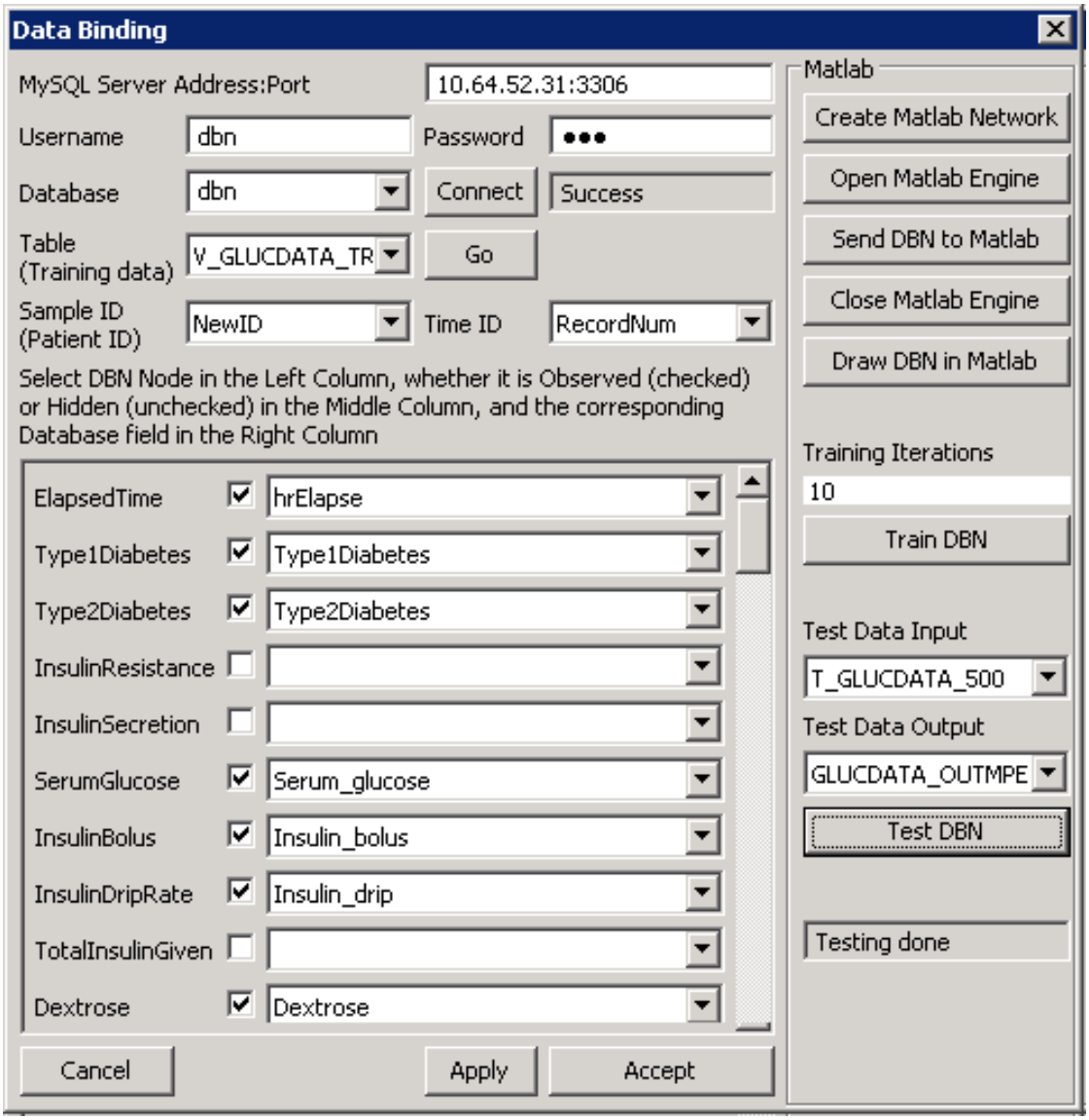


Figure 3.3: Projeny data binding dialog

the table with the training data, and then the database columns with the patient identifier and the timeslice identifier. After this is done, the user checks the observed nodes in the pane at the bottom left, and chooses the database column that contains the data for these observed nodes from the pick list, which is prepopulated with the names of all the database columns in the table. The user leaves the check boxes unchecked for hidden nodes, and does not select a database column to provide the training or testing data for these nodes.

3.2.4 Training the Model (Parameter Learning)

Projeny uses the two-timeslice DBN EM learning algorithm implemented in BNT to learn the parameters of the model. The Expectation Maximization algorithm imputes missing values in the training data set, and initializes the parameter learning algorithms with the most likely data set given the model. The learning algorithms calculate parameters for each equivalence class and store the results in d-dimensional conditional probability tables. Here, $d = 1 + numparents$, where numparents is the number of parents of a given node in the DBN. These probability tables can be saved to the computer's file system as an XML file or a Matlab workspace file using specific Matlab commands. They can be loaded into the Matlab engine in the future so that training need not be repeated.

3.2.5 Testing the Model

After training is completed, three tables are selected for testing the model. These three tables include one input table with the test data set, and two output tables - one for the most probable estimate, and one for the marginal probability distribution for each node that is estimated using the inference algorithms. Currently, Projeny uses a junction-tree-based DBN inference algorithm implemented in BNT[42]. Other inference algorithms are not supported at present.

After selecting the input and two output tables, the inference algorithm is run by clicking the 'Test' button. The junction tree algorithm calculates the marginal probability distributions for any node whose data were not provided in the test data set. This allows the model to estimate the values of all the nodes that were not known in a single step. This is an advantage of the DBN model over models such as regression and neural networks that have preset input and output variables. The DBN model takes all the nodes whose

values are provided as the input data for a specific patient, and estimates the marginal distributions of all the nodes whose values were left blank, and saves these to the output table.

After completing the testing process, the expected value of a given node may be calculated from the marginal distributions using the formula,

$$EV = \sum x_i \, p_i \tag{3.1}$$

where $x_i$is the midpoint of the interval of the i^{th}state of the given node, and $p_i$is the calculated probability associated with the i^{th} state of this node, and is obtained from the marginal probability distribution. EV is the expected value of the node of interest in a timeslice of interest.

A common mode of testing the accuracy of the model is to hide variables of interest at random from the test data set, and then estimating their value using the model. The model is made to calculate the probability distributions of the variables of interest. The expected value of these variables can then be calculated from the probability distributions. The actual and the estimated values can then be compared using statistical tests. This technique is used in both the insulin-glucose models and sepsis models in our experiments. Evaluation using this and other evaluation methods are presented in Chapters 4 and 5.

CHAPTER 4

TEST CASE 1: STRESS-INDUCED HYPERGLYCEMIA IN THE ICU

Hyperglycemia (increase in blood glucose or serum glucose above the normal range) is commonly seen in patients who are critically ill, and are hence treated in the Intensive Care Units (ICU). Intensive insulin therapy to maintain serum glucose within normal levels has been shown to improve patient outcomes. We modeled glucose homeostasis and insulin dosing in the ICU using Dynamic Bayesian Networks. This chapter discusses the models, the evaluation of their accuracy, and evaluation of their safety and efficacy compared to a protocol that is currently being used to control serum glucose levels of patients in the ICU.

4.1 Overview

Hyperglycemia, in patients who are critically ill, is considered to be a response to stress, due to chemical mediators such as catecholamines and glucocorticoids that are released by the body in response to acute stress caused by the illness, and due to drugs such as steroids, diuretics, and some antiviral drugs which are administered as part of the treatment. Hyperglycemia is seen in patients who are critically ill, including those who are not diabetic, due to these stress response mechanisms[33].

4.1.1 Overview of the Problems

Hyperglycemia was once considered to be part of the body's normal response to stress. However, several recent studies have shown that increased serum glucose levels further worsen the patient's condition and the outcomes, whereas active control of serum glucose levels and maintaining them within the normal range improves outcomes, and hence is desired. The older treatment to achieve glucose control is to administer insulin when the serum glucose is greater than 215 mg/dL[158]. A relatively newer intensive insulin

therapy aims to maintain the patients' serum glucose between 80 and 110 mg/dL by actively adjusting the intravenous glucose drip and intravenous insulin dosage with periodic monitoring and adjustment once every 2 hours or less[159].

In clinical trials involving critically ill patients, those who had intensive insulin therapy demonstrated reduced mortality and morbidity compared with patients who had conventional insulin therapy[159]. However, a few studies have reported that patients undergoing intensive insulin therapy have slightly increased mortality due to hypoglycemia[160]. The usefulness of intensive insulin therapy is currently debated by the medical community[160], but the general consensus is that it reduces mortality and morbidity of critically ill patients[34], especially of those in surgical intensive care[160][34].

Patients in the ICU are under continuous monitoring by clinicians and their physiological parameters are measured at regular, short intervals. Considering the severe mortality and morbidity associated with hyperglycemia during critical illness, the improved outcomes that can be achieved with good glucose control, and the rich availability of data in the ICU, this problem lends itself well to computerized clinical care protocols. Morris states that a clinical protocol must be 'adequately explicit' to reduce interclinician variability during its application[161]. We posit that an adequately explicit protocol will provide sufficient information to be implemented in a computer system and execute without human intervention to find a solution. However, due to various combinations of comorbidities and their interactions, human interpretation and intervention is almost always required[162][163]. Hence, a decision support system that helps the clinicians to understand the patient's condition easily and make better decisions is still highly useful[162].

Several computerized and paper-based models have been used to monitor the serum glucose and calculate the insulin dose. Two popular rule-based protocols are the Yale Insulin Infusion Protocol[164], a paper-based protocol developed at Yale University, and eProtocol-insulin[37], a computerized rule-based protocol developed at LDS Hospital in Salt Lake City.

4.1.2 Overview of the Experiments

We chose to model insulin dosing and glucose homeostasis in cases of stress induced hyperglycemia in the ICU using a temporal probabilistic model. The first objective of our experiments is to test whether our models can predict the serum glucose and insulin dose accurately in retrospective data using partial information of the patients in the test data set. This task is described as Viterbi Decoding in Section 2.5.1. Viterbi decoding is one of the important inference tasks for the DBN, which estimates the values of the hidden (or unknown) nodes given the values of observed (or known) nodes from a given point in time in the past until the current point in time. In this dissertation, we call these experiments 'Insulin and Glucose Estimation' experiments, or simply 'Estimation' experiments.

After proving that the model can predict the past sequence of serum glucose and insulin drip rates accurately, we wanted to test the ability of the model to control the serum glucose of the patient within a normal range at a future point in time, by recommending an appropriate insulin dose at the current point in time. We perform the fixed-lag smoothing task described in Section 2.5.1 to recommend an insulin dose at a given point in time to bring the serum glucose within normal range at a future point in time. The glucose dose recommended by the DBN model was compared to the glucose dose recommended by eProtocol-insulin using a technique described by Wong et al.[165] In this dissertation, we call these models and experiments 'Insulin Dose Recommendation' experiments, or simply 'Recommendation' experiments.

Both the Estimation and Recommendation experiments were performed with two different discretization techniques each. The two discretization techniques we used are a combination of domain-based and equal interval discretization, and k-means clustering. Table 4.1 lists the different discretization techniques and experiments performed with the 'stress-induced hyperglycemia in the ICU' test case.

4.1.3 Data Set for the Experiments

The data for this test case were a subset of the eProtocol-insulin data of adult patients collected during a 2-year period from January 2004 to December 2005 at LDS Hospital. Institutional Review Board (IRB) approval was obtained from University of Utah and Intermountain Healthcare. The anonymized data set was provided by Kathy Sward, and had

Table 4.1: Stress-induced hyperglycemia in the ICU - experiments

	Discretization technique	
Objective of Experiment	Domain-based + equal interval	k-means clustering
1. Glucose and Insulin Estimation	Experiment 1	Experiment 2
2. Insulin Dose Recommendation	Experiment 3	Experiment 4

been used for a previous research project on computerized insulin dosing protocols[166]. The data set contained the data of patients treated in the Shock, Trauma, Respiratory Intensive Care Unit (STRICU) at LDS Hospital. Patients with less than three timeslices had their data excluded from the training and test data sets, since Projeny requires at least three timeslices of data for each patient. Temporal data of 796 patients were included for the experiments.

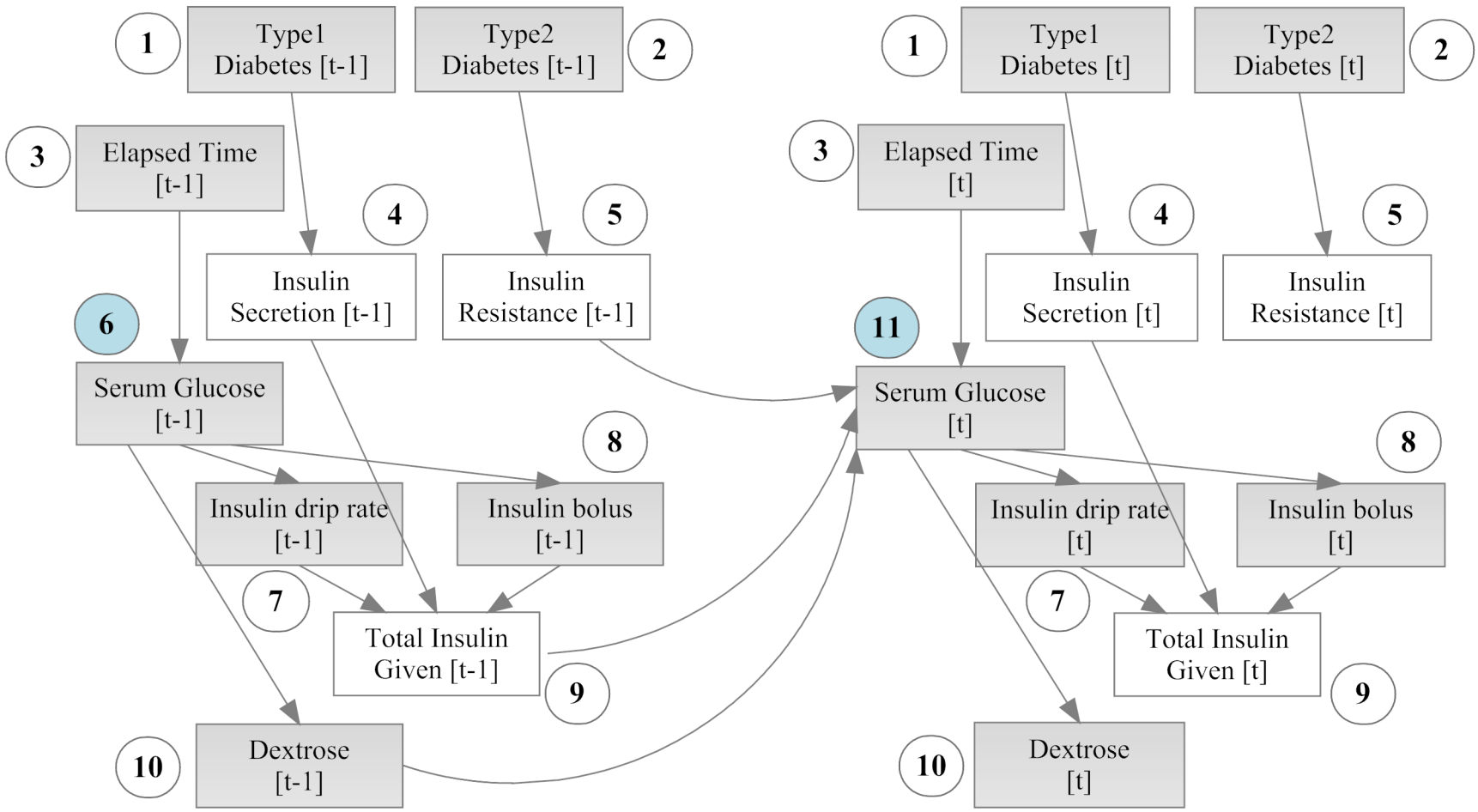
The data were then temporally aggregated and consolidated as described in Section 3.1.1. Variables listed in Table 4.2 were included for the four experiments. They were discretized using the two techniques listed in Table 4.1, and are described in further detail in context of each experiment in this chapter.

4.1.4 Model Structure

The model structure was derived using domain knowledge from clinical literature. The structure of the model is shown in Figure 4.1. The patient's type 1 and type 2 diabetes mellitus statuses were included in the model, as they determine the patient's insulin secretion

Table 4.2: Data elements selected for hyperglycemia experiments

Variable description	Acronym
Anonymized patient identifier	FakeMRN
Timestamp	Timestamp
Serum Glucose measurement	SerGluc
Dextrose IV bolus dose	Dextrose
Insulin bolus dose (in U)	InsulinBolus
Insulin drip rate (in Units/hr)	InsulinDrip
Patient's Type 1 diabetes mellitus status	Type1DM
Patient's Type 2 diabetes mellitus status	Type2DM



Legend: Rectangular nodes are discrete. Edges show intraslice (straight lines), and interslice (curved lines) connections. Clear nodes are hidden. Shaded nodes are observed. Numbers in circles indicate equivalence classes. All pairs of nodes except serum glucose are in the same equivalence class in the two timeslices.

Figure 4.1: Two-timeslice model of hyperglycemia DBN

and sensitivity to a certain extent even in patients who are hyperglycemic due to severe illness. We modeled the patient's insulin secretion and insulin resistance as hidden nodes, with type 1 and type 2 diabetes mellitus statuses as their respective intraslice parents. They cannot be measured directly in a patient, but they can be estimated from other observed variables. We made an intraslice connection from insulin secretion to total insulin given. The node 'total insulin given' represents the total amount of insulin introduced into the patient's bloodstream in a given timeslice. It is composed of endogenous insulin secretion, insulin bolus dose, and intravenous (IV) insulin drip. Since endogenous insulin secretion is a hidden variable, total insulin given is also modeled as a hidden variable, with the three aforementioned parents. Insulin bolus dose and insulin IV drip rate are observed nodes. Intraslice connections are made between serum glucose and insulin bolus dose, and between serum glucose and insulin drip rate, because the insulin bolus doses are determined by the serum glucose level. An intraslice connection was also made from serum glucose to dextrose IV bolus dose. Interslice connections were made to serum glucose in the second timeslice, from three nodes - dextrose, total insulin given, and insulin resistance, in the first timeslice. All the nodes were modeled as discrete variables. Time was modeled as discrete as well, using timeslices that were 2 hours wide. The choice of 2-hour wide timeslices was made to support the eProtocol-insulin data which contained data that were measured once every 2 hours.

In Figure 4.1, square or rectangular nodes indicate discrete variables. Continuous variables, which are indicated by circular nodes by convention, are not included in this model. Clear nodes indicate hidden variables and shaded nodes indicate observed variables. The figure shows a two-timeslice Dynamic Bayesian Network with intraslice and interslice connections. The equivalence classes are identified with numbers next to each node. The two timeslice model has 11 equivalence classes, as described in Section 3.2.2. The parameters are tied between nodes in different timeslices in the same equivalence class. In our data set, all patients had more than two timeslices. In such cases, the model will be unrolled to as many timeslices as are found in the patient's data by the learning and inference algorithms used. The equivalence class of a given node will be the same from the second until the last timeslice. For some nodes, the equivalence classes will be the same from the first until the last timeslice, if their cohorts in the first and second timeslices

are in the same equivalence class. As described in Section 3.2.2, this reduces the number of parameters required to describe the model by a factor of t, where t is the number of timeslices in a given patient's data set.

All the models used in the four experiments in this chapter have the same nodes and edges. However, the states of the nodes are different depending on the discretization technique used. The learning and inference questions, as well as the data preparation steps, are different. These are described in the following sections.

4.2 Objective 1: Glucose and Insulin Estimation

The first task we performed to validate our methods, tools, and techniques was to confirm that our learning and algorithms work with good accuracy. In other words, we hypothesized that our models can estimate a given patient's historical values of serum glucose and insulin drip rate with good accuracy. For the two experiments performed to validate this objective, we hypothesize that there is no significant difference between the actual and the predicted historical values of both serum glucose and insulin drip rate.

We divided the data at random with two-thirds of the patients assigned to the training data set, and the remaining one-third of patients assigned to the test data set. The model was first discretized using a combination of domain-based and equal interval discretization. The model did not show very high accuracy, as described in Section 4.2.1. Hence, we tried k-means discretization as an alternative discretization approach, which is described in Section 4.2.2.

4.2.1 Experiment 1: Domain-based and Equal interval Discretization

The objective of this experiment was to test whether the DBN model can accurately predict the historical values of serum glucose and insulin drip rate, using a data set prepared using a combination of domain-based and equal interval discretization.

4.2.1.1 Materials and Methods

4.2.1.1.1 Data preparation. The normal range for several physiological variables is described in medical literature. It is possible to define the normal, high, very high, low,

and very low ranges for serum glucose using clinical literature or by interviewing the clinical domain experts. We used clinical literature to derive these values. After this was performed, the states for serum glucose for the DBN model were defined. To approximate the clinical decision making process by human experts, the width of the intervals were narrow in the normal range for serum glucose, and were progressively widened as the serum glucose value moved further away from the normal range. These are illustrated in Table 3.2.

Elapsed Time was discretized using equal interval discretization, with a bin width of 2 hours. Insulin drip rate could not be stratified into low, high, and normal ranges from clinical literature. Hence, it was discretized using equal interval discretization with a bin width of 2 U/hour. Both type 1 and type 2 diabetes mellitus were binary nodes, with two states: absent and present. Both insulin secretion and insulin resistance were binary nodes as well, with two states: normal and impaired. Dextrose dose was an ordinal variable, which denoted various bolus doses of glucose administered intravenously. We did not have access to the patients' severity of illness, nutrition (enteral and parenteral feeding), or medication information. Hence, these variables were not included in the model, even though they were very relevant.

4.2.1.1.2 Learning. The training data set had the data of 508 patients, having 3 to 209 timeslices each. Expectation-Maximization-based parameter learning was used to learn the parameters, with a maximum number of iterations set to 100. The EM algorithm converged after 19 iterations and took about 5 hours. We used a computer with a 3GHz quad-core Intel Xeon processor with 8GB of RAM, running Matlab and Projeny. The database was hosted on a separate server. Figure 4.2 shows the log likelihood for training for the first 10 iterations, showing the Expectation Maximization algorithm getting close to convergence.

4.2.1.1.3 Inference. The test data set had the data of 287 patients, having 3 to 366 timeslices each. Testing was done by randomly removing 20% of observed values of serum glucose and insulin drip rate. If a patient had a data set of t timeslices, $t/5$ instances each of serum glucose and insulin drip rate were removed by changing their values to null values in the database. This process was done for all the patients in the test data set. These removed values were then estimated by using a junction tree-based exact inference

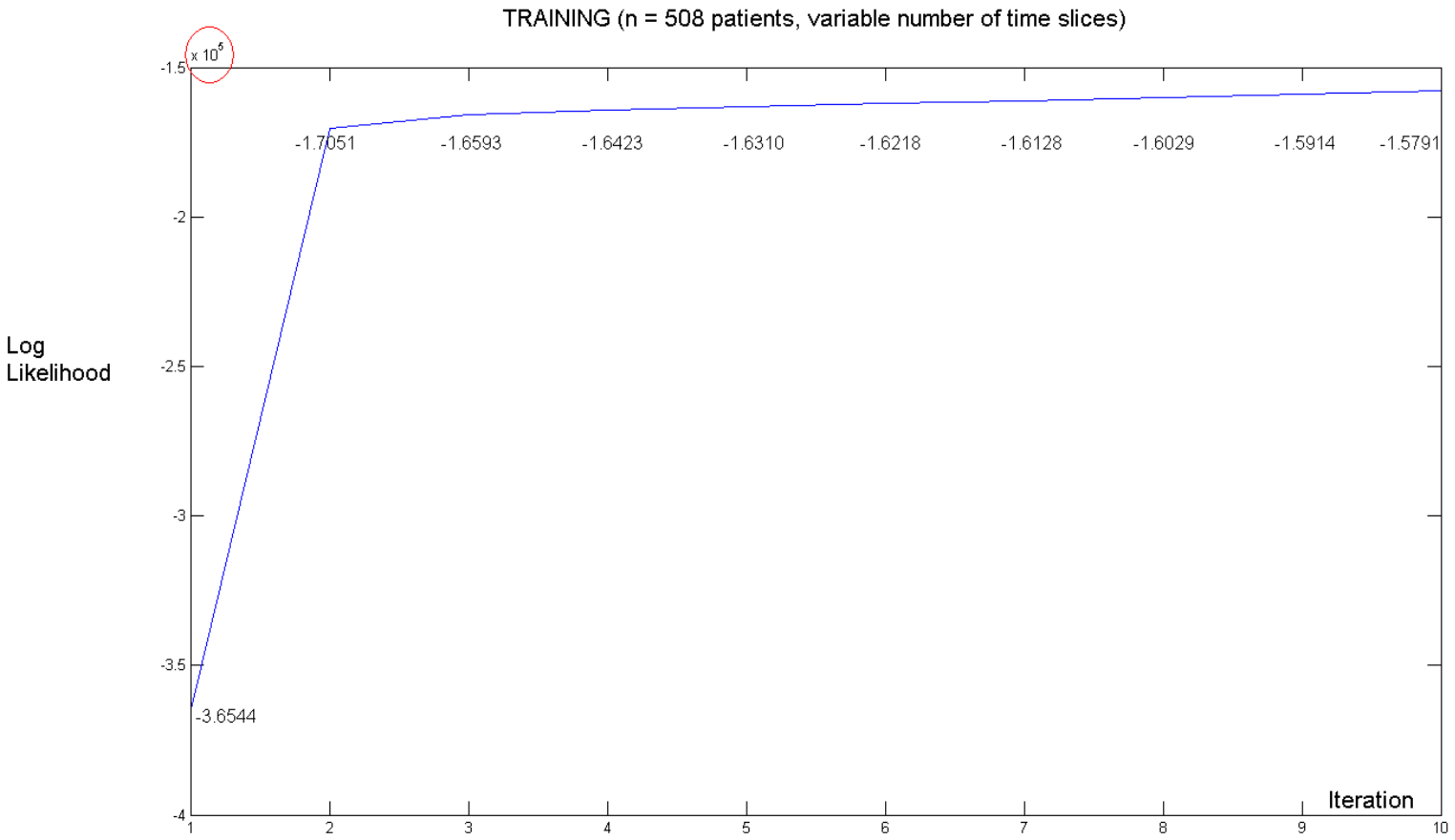


Figure 4.2: Log-likelihood over 10 iterations of training using EM algorithm

algorithm. The inference algorithm calculated marginal probability distributions for all the values that were null in the input data. The inference algorithm took about 1 hour to complete. From these probability distributions, the expected values and their standard deviations were calculated for these nodes of interest, using the procedure described in Section 3.2.5.

4.2.1.2 Results

The model was validated by comparing the expected values estimated by the inference algorithm for randomly removed serum glucose and insulin drip rate values against their actual values. Correlation coefficients were calculated. Serum glucose was normally distributed in both training and test data sets. Hence, Pearson correlation coefficient is a valid measure for this variable. However, insulin drip rate had a skewed distribution, and hence, Pearson correlation coefficient is not applicable. Spearman correlation coefficient was calculated for insulin drip rate. The correlation coefficients are shown in Table 4.3.

The results from this experiment show that the model had very high correlation for serum glucose and high correlation for insulin drip rate. We also plotted the difference between the actual and predicted values of serum glucose and insulin drip rate in terms of number of states, as shown in Figures 4.3 and 4.4. These figures show the high accuracy of the model in estimating the clinical course of these patients over time.

Figures 4.3, and 4.4 plot the number of states of difference between actual and predicted values on the X-axis. The Y-axis represents the number of cases whose actual and predicted values differed by the number of states on the X-axis. These two figures show that the predicted values were within three states of the actual values in about 95% of cases

Table 4.3: Hyperglycemia experiment 1. Correlation coefficients.

Serum glucose estimation	
Pearson's correlation coefficient	0.810699
Spearman's correlation coefficient	0.997237
Insulin drip rate estimation	
Pearson's correlation coefficient	*N/A**
Spearman's correlation coefficient	0.999955
*Pearson's correlation coefficient was not considered for Insulin Drip rate because it was not normally distributed in training or testing data sets	

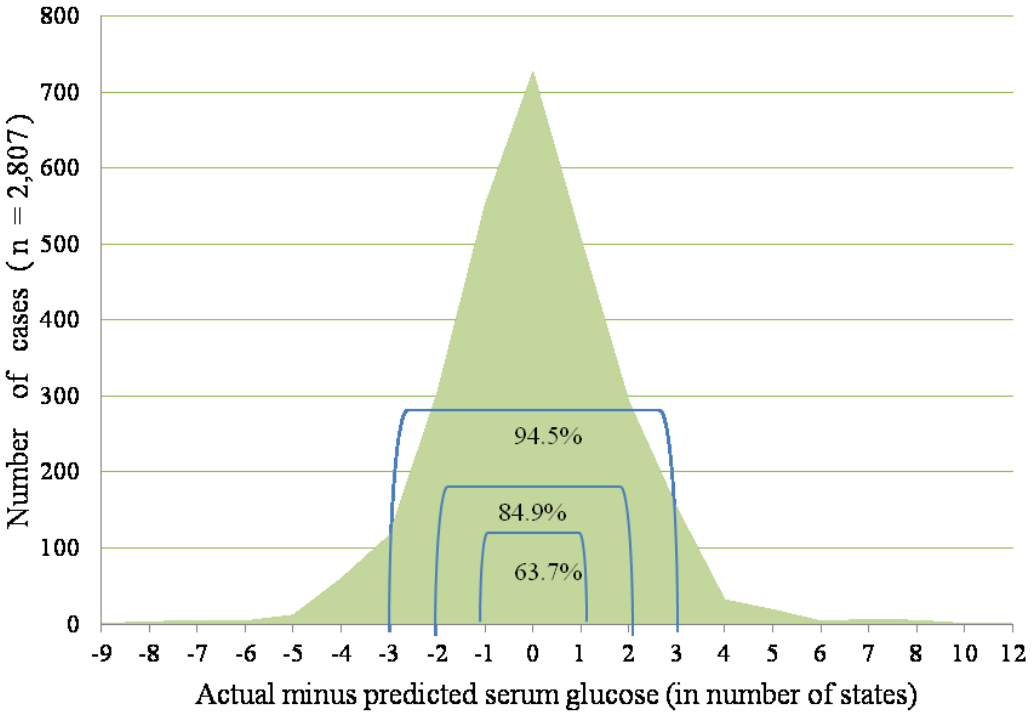


Figure 4.3: Hyperglycemia experiment 1. Serum glucose, actual vs. predicted

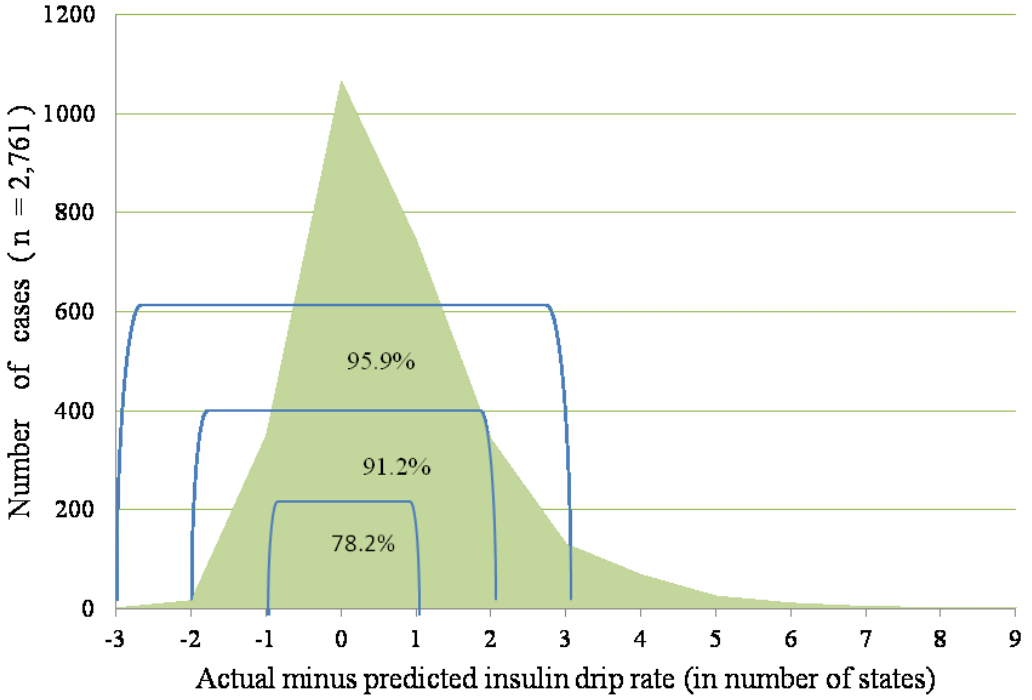


Figure 4.4: Hyperglycemia experiment 1. Insulin drip rate, actual vs. predicted

for both serum glucose and insulin drip rate. However, of more interest is the finding that the serum glucose was predicted within 1 state of the actual value in 63.7% of cases, and the insulin drip rate was predicted within 1 state of the actual value in 78.2% of cases, both showing a high accuracy.

4.2.2 Experiment 2: K-means Clustering

We wanted to test whether the accuracy of the model can be improved by using a discretization technique that preserved the information content better than the domain-based or equal interval discretization techniques. Hence, we selected the k-means clustering technique, which discretizes the input data into a specified number of bins based on the inherent clustering observed in the data. This technique is described in detail in Chapter 3.

4.2.2.1 Materials and Methods

The nodes and the edges in the model were kept the same as in Figure 4.1. However, we discretized all the continuous observed variables in the model using k-means clustering, with k set to 10 and 15. We used the discretization produced with k = 15 if the error rate was much less than with k = 10. If there was not a significant reduction in error rate, we used k = 10 bins. Learning and inference were performed as before, as mentioned in Section 4.2.1, using the same training and testing populations, albeit with different discrete states due to the change in the discretization techniques. Expected values were calculated using the same technique as in experiment 1.

4.2.2.2 Results

The inference algorithm calculated the *marginal probability distributions (MPD)*, which are the probabilities for various states of the estimated variables. The inference algorithm also calculated the most likely state of each variable, known as its *most probable estimate (MPE)*, based on its marginal probability distribution. The most probable estimates for all estimated variables of a patient chosen at random are shown in Figure 4.5. The variable names shown at the top row in the figure correspond to the rows in the matrices shown in the figure. The first matrix named *tevidence* shows the input data of the patient, with each row representing a node in the model, and each column representing a timeslice. The

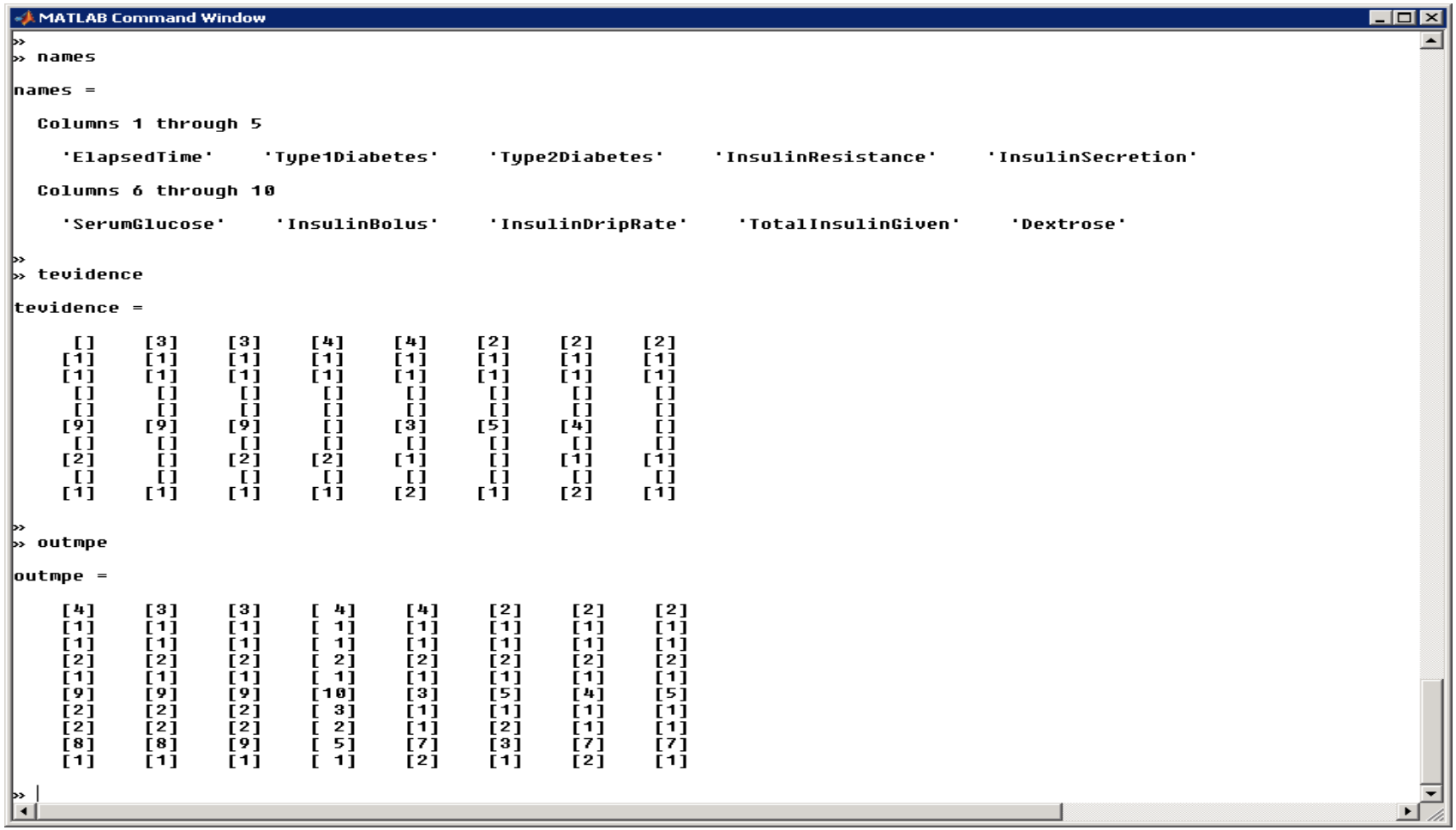


Figure 4.5: Hyperglycemia experiment 2. MPEs from the DBN model

number in each cell in the matrix represents the discrete state of the specific node in the specific timeslice. It must be noted that the states are numbered beginning at 1 (i.e., they are not zero-indexed). The cells that are blank denote either hidden nodes or missing data. The missing data may either denote true missing data, or the data removed at random for estimation by the model as part of our experiment.

The matrix named *outmpe* in the figure denotes the results at the end of the inference. The values of all the blank nodes were estimated by the inference algorithm. The figure shows the most probable estimates.

We calculated the expected values and their standard distributions from the marginal probability distributions, and we used these two measures for our statistical analyses. The correlation coefficients for this experiment are shown in Table 4.4.

A comparison between Tables 4.3, and 4.4 shows that k-means clustering produced more accurate results than a combination of domain-based and equal interval discretization. This difference is more pronounced in the case of the second objective: recommending insulin dose to maintain a normal serum glucose level, described in Section 4.3.

Figures 4.6 and 4.7 show the difference between the actual and predicted serum glucose and insulin drip rates, respectively, in terms of number of standard deviations. The mean and standard deviation of the actual serum glucose values were 110.6 mg/dL and 36.31 mg/dL, respectively. The mean and standard deviation of predicted serum glucose values were 111.992mg/dL and 29.822 mg/dL, respectively. The mean and standard deviation of the actual insulin drip rate values were 3.457 U/hour and 2.702 U/hour, respectively. The mean and standard deviation of predicted insulin drip rate values were 3.415 U/hour and 0.735 U/hour, respectively.

Table 4.4: Hyperglycemia experiment 2. Correlation coefficients.

Serum glucose estimation	
Pearson's correlation coefficient	0.831984
Spearman's correlation coefficient	0.996909
Insulin drip rate estimation	
Pearson's correlation coefficient	*N/A**
Spearman's correlation coefficient	0.999995
*Pearson's correlation coefficient was not considered for Insulin Drip rate because it was not normally distributed in training or testing data sets	

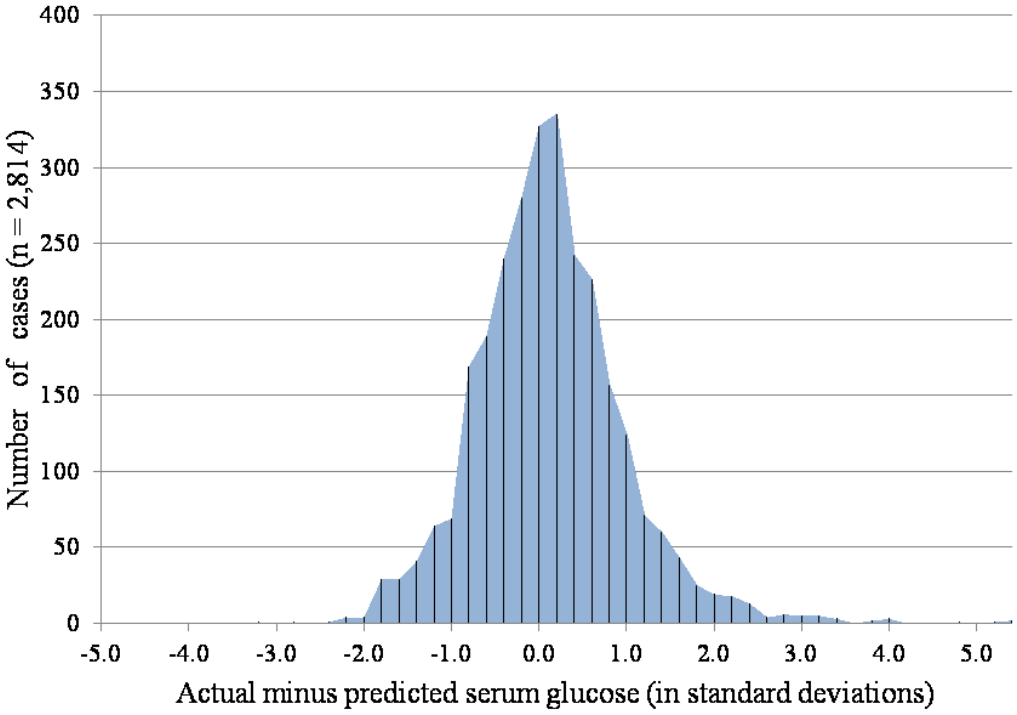


Figure 4.6: Hyperglycemia experiment 2. Serum glucose, actual vs. predicted

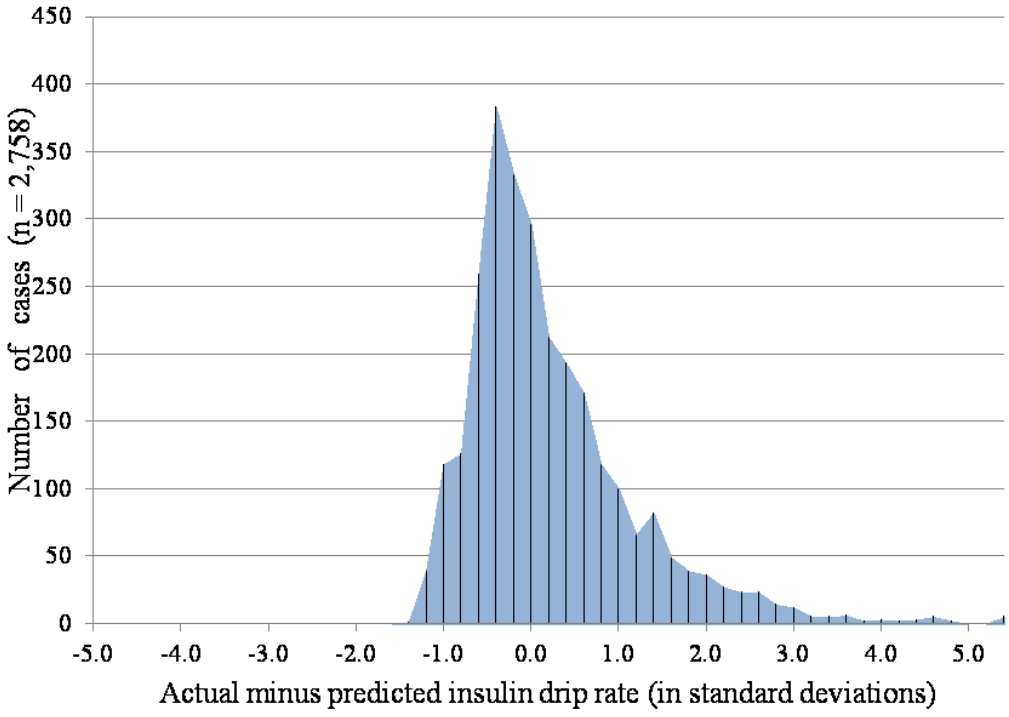


Figure 4.7: Hyperglycemia experiment 2. Insulin drip rate, actual vs. predicted

Figures 4.6 and 4.7 show that a vast majority of predicted values of serum glucose and insulin drip rate are within one standard deviation of the actual values. The figures show that the model was highly accurate in predicting the serum glucose and insulin drip rate even in the absence of data about the patients' nutrition and severity of illness.

4.3 Objective 2: Recommendation of Insulin Drip Rate for Glucose Control

The first objective of the hyperglycemia test case was to validate that the model works according to its theoretical underpinnings, and predicts variables in the model with accuracy using real-world clinical data. After proving this objective, our next objective was to prove the clinical validity of the model in a real-world clinical use as well. A rule-based protocol known as eProtocol-insulin is currently used on bedside computer terminals in the Shock Trauma Respiratory ICU of LDS Hospital. eProtocol-insulin is an open-loop computerized clinical decision support system that allows the clinical expert to enter the values of the current serum glucose and nutrition information. eProtocol-insulin then computes the next insulin drip rate using a rule-based protocol, and recommends this dosage to the clinical expert. The clinician may accept or reject this recommendation, and enter the action taken. If the recommendation is rejected, the clinician may enter the reason for rejecting it, and the new insulin drip rate that is administered to the patient. If the insulin drip rate is modified manually by the clinician, it is entered into the computer program.

The data set used for our experiments was also created by the eProtocol-insulin computer application. Hence, we considered eProtocol-insulin as the gold standard against which to compare the DBN model. Comparison was performed using a technique developed by Wong et al.[165]

4.3.1 Evaluation Technique

The goal of the experiments is to compare both the efficacy and safety of the insulin dose recommended by two different protocols to maintain serum glucose within the normal range. eProtocol-insulin has a target insulin dose of 95 mg/dL. At any given point in time, the future insulin drip is calculated taking into account the current drip rate, current

serum glucose level, and current nutrition information to achieve a future serum glucose of 95mg/dL. The evaluation technique took actual data from eProtocol-insulin, and compared it to simulated data from the new protocol or model that is being evaluated (e.g., DBN model).

Each test-case for this evaluation technique is a four timeslice model. Hence, patients with less than four timeslices of data were excluded from this evaluation. Test cases were constructed by selecting every consecutive four timeslice data set from current patients using a moving window which was four timeslices wide. For example, if a patient had a data set six timeslices in length, three test cases were constructed by selecting first to fourth, second to fifth, and third to sixth timeslices. If a patient had t timeslices of data where $t > 4$, then $t + 1 - 4$ test cases can be created. We created 10,433 test cases from the temporal data of 287 patients. These 10,433 test cases provided a large sample size for our test data set to compare the efficacy and safety of the DBN model's insulin drip rate recommendation with that of eProtocol-insulin.

The technique is explained using Table 4.5. We took each 4-timeslice test case, and used the first three serum glucose measurements (G_1, G_2, G_3) as-is. We changed the fourth serum glucose (G_4) value to 95 mg/dL. We provided the first two insulin drip rates (I_1, I_2) to the model, as well as the time elapsed (T_2, T_3, T_4) between any two timeslices. The DBN model is now made to predict the insulin drip I_3 rate at time T_3, to achieve a serum glucose level G_4 of 95 mg/dL at time T_4. The insulin drip rate $G_{3_{eProt}}$ recommended by eProtocol-insulin at time T_3 is already available in our data set. The two insulin drip rate recommendations, the predicted dose recommendation G_3 by DBN model, and the actual real-world dose recommendation $G_{3_{eProt}}$ by eProtocol-insulin are now compared.

The model that predicted a better insulin dose in each of the 10,433 test cases is considered the winning model in that case. The word 'better' is defined in terms of safety and efficacy. If the actual observed serum glucose G_4 at time T_4 was greater than 95 mg/dL, the model (eProtocol-insulin or DBN model) that recommended the higher insulin drip rate I_3 at time T_3 is considered to have better efficacy. For example, if the serum glucose at time T_4 was 140 mg/dL, and if the DBN model recommended a higher insulin drip rate at time T_3 than the recommendation of eProtocol-insulin, then DBN is the winning model for this test case. If the DBN model recommended a lower insulin

Table 4.5: Evaluation of DBN protocol against eProtocol-insulin

Variable	T1	T2	T3 (time of interest)	T4 (in the future)
Serum glucose (G_x)	Observed	Observed	Observed	Set to a target of 95
Insulin drip rate (I_x)	Observed	Observed	*Predicted by DBN or eProtocol-insulin*	Not known
Time lapse (IL_x)	Not applicable	Observed	Observed	Known

dose than eProtocol-insulin, then eProtocol-insulin is the winner in this case, because serum glucose G_4 was higher than the target range. Hence, the necessary dose to control the glucose adequately must be larger. Even if both models did not control the glucose adequately, the model that predicted the higher glucose dose is the winner in this case.

Conversely, safety is measured in terms of whether the model produced hypoglycemia during time T_4. Hypoglycemia (serum glucose below the normal range) is a serious life threatening condition, and it is safer to have mild hyperglycemia rather than hypoglycemia. If the patient's serum glucose at time T_4 is below 95 mg/dL, then the model that recommended the lesser insulin drip rate at time T_3 is the safer one and is considered the winner in this case. Ties were not encountered in experiments 3 and 4.

We performed this evaluation with two different data sets created by using the two different discretization techniques as described in objective 1.

4.3.2 Experiment 3: Domain-based and Equal Interval Discretization

We measured the efficacy and safety of the insulin drip rate recommendations of the model and data set that we used in experiment 1, by applying the technique described in Section 4.3.1.

4.3.2.1 Materials and Methods

The training and the testing model structures, the states of the discrete variables, and the training data set were the same as in experiment 1, which used a combination of domain-based and equal interval discretization. The test data set was discretized using

the same technique, a combination of domain-based and equal interval discretization as described in experiment 1. However, each test case itself was a four-timeslice data set created using the procedure described in Section 4.3.1.

4.3.2.2 Results

We stratified the test results by the different ranges of serum glucose in timeslice T_4. We then counted the number of cases where the DBN model gave a higher dose than eProtocol-insulin and vice versa, which resulted in serum glucose at time T4 falling within this range. The results are shown in Table 4.6.

This experiment shows DBN performed better than eProtocol-insulin in a large number of cases where G4 showed moderate hyperglycemia. However, eProtocol-insulin performed better than DBN model in a large number of cases where G4 showed severe hyperglycemia or severe hypoglycemia. DBN model recommended insulin drip rates that were drastically different from those recommended in the real world by eProtocol-insulin. This may be explained by the way the data were modeled in the case of the DBN model, leading to overfitting in the more common cases, but insufficient accuracy in the less en-

Table 4.6: Hyperglycemia experiment 3, eProtocol-insulin vs. DBN model

Actual G4	DBN-I3 > eProt-I3 (Num of cases)	DBN-I3 < eProt-I3 (Num of cases)	Winning model	Winning margin
>500	0	0	-	-
400 - 500	1	0	-	insignificant
300 - 400	3	10	eProtocol	54%
200 - 300	33	56	eProtocol	26%
150 - 200	235	168	DBN	17%
110 - 150	1612	722	DBN	38%
100 - 110	1016	399	DBN	44%
90 - 100	1025	426	DBN	41%
80 - 90	940	356	eProtocol	45%
70 - 80	570	188	eProtocol	50%
55 - 70	236	102	eProtocol	40%
40 - 55	27	13	eProtocol	35%
<40	0	1	-	insignificant

countered cases. Domain-based discretization combined with equal interval discretization does not seem adequate for accurate training and testing in this test case.

4.3.3 Experiment 4: K-means Clustering

We wanted to improve the accuracy of predictions performed by the DBN model. We tested whether the accuracy can be improved by using a discretization technique that preserves the information content of the data better than domain-based and equal interval discretization techniques. Therefore, we measured the efficacy and safety of the insulin drip rate recommendations of the model and data set that we used in experiment 2, by applying the technique described in Section 4.3.1.

4.3.3.1 Materials and Methods

The training and the testing model structures, the states of the discrete variables, and the training data set were the same as in experiment 2, which discretized the data using the k-means clustering technique. The test data set was discretized using the same technique as described in experiment 2. However, each test case was a four-timeslice data set created using the procedure described in Section 4.3.1.

4.3.3.2 Results

We again stratified the test results by the different ranges of serum glucose in timeslice T_4. These ranges were defined by Wong et al. in their description of the technique. We then counted the number of cases where the DBN model gave a higher I_3 dose than eProtocol-insulin and vice versa, which resulted in serum glucose G_4 (measured at time T_4) falling within this range. The results are shown in Table 4.7.

The glucose ranges in this table are identical to the same ranges used to evaluate the results in experiment 3. Hence, a direct comparison can be performed between the difference due to the two discretization techniques. The current experiment shows that both eProtocol-insulin performed equally well. eProtocol-insulin was more efficacious in cases of severe hyperglycemia and safer in moderate hypoglycemia.

However, the DBN model was more efficacious in moderate hyperglycemia, and safer in severe hypoglycemia. This behavior of the DBN model is a desirable quality, since

Table 4.7: Hyperglycemia experiment 4, eProtocol-insulin vs. DBN model

Actual G4 (mg/dL)	DBN-I3 > eProt-I3 (Num of cases)	DBN-I3 < eProt-I3 (Num of cases)	Winning model	Winning margin
>500	0	0	-	-
400 - 500	1	0	DBN	insignificant
300 - 400	5	8	eProtocol	23%
200 - 300	40	49	eProtocol	10%
150 - 200	203	197	DBN	2%
110 - 150	1340	965	DBN	16%
100 - 110	708	583	DBN	10%
90 - 100	706	599	DBN	8%
80 - 90	644	509	eProtocol	12%
70 - 80	382	269	eProtocol	17%
55 - 70	133	164	DBN	10%
40 - 55	15	22	DBN	19%
<40	1	0	eProtocol	insignificant

avoidance of severe hypoglycemia is more important than controlling moderate hyperglycemia. The difference in number of cases in each of the range intervals expressed as a percentage, also known as winning margin, ranged between 2%, and 23%. There was not an extreme difference between the two models. However, the DBN model was more efficacious and safer in a larger number of cases than eProtocol-insulin.

4.3.4 Discussion

Experiments 1 and 2 show that the DBN model estimated the serum glucose levels and the insulin drip rates with high accuracy. Between these two experiments, k-means clustering provided a more accurate estimation of serum glucose levels and insulin drip rates. Experiments 3 and 4 show that k-means discretization provides more accurate inference, and by extension, parameter learning, compared to a combination of domain-based and equal interval discretization.

Considering the four experiments together, it can be seen that k-means clustering performed better than the manual discretization models. This can be attributed to the fact that k-means clustering takes into account natural groupings of data points in the sample

whereas manual discretization does not. This leads to better estimation of parameters during training, and hence more accurate results from the test iterations.

The DBN model performed better and safer than eProtocol-insulin in most cases, even though the DBN model did not have access to the patients' feeding data, whereas eProtocol-insulin has these data available. The accuracy of the DBN can be improved by using a more complete data set containing the patients' nutrition and severity of illness data. The DBN model and k-means discretization may be validated with a more complete data set for their use in a real-world clinical setting.

Statistical analysis for experiments 1 and 2 were done by calculating the Pearson and Spearman correlation coefficients. However, statistical analysis was not performed in experiments 3 and 4, since they were qualitative comparisons of safety and efficacy. Statistical measures such as the Youden's J index (the maximum value of sensitivity + specificity - 1)[167], and other ROC curve comparison methods[168] are available for comparing the accuracy of two diagnostic tests that determine the presence or absence of a disease. However, both experiments 3 and 4 involved qualitative comparisons of safety and accuracy stratified by serum glucose ranges. A statistical test is not available to compare qualitative estimates of safety and efficacy of two prediction algorithms. Hence, a quantitative comparison of the safety and efficacy of the two protocols could not be performed in experiments 3 and 4.

CHAPTER 5

TEST CASE 2: EARLY DETECTION OF SEPSIS IN THE ED

Sepsis is defined as a combination of Systemic Inflammatory Response Syndrome (SIRS, a state of increased immune response), and a confirmed or suspected infection, usually caused by bacteria. The infection may be present in lungs, skin, urinary tract, bone, brain and meninges, intestines, blood, or other tissues. Sepsis may occur whether the infection stays localized or spreads to other parts of the body. Presence of bacteria in blood, known as bacteremia, does not by itself denote sepsis in the absence of a systemic inflammatory response. Untreated or inadequately treated cases of sepsis can lead to a condition known as severe sepsis, which is characterized by complications such as an uncontrollable fall in blood pressure, hemodynamic collapse, multiple organ failure, or death. Early diagnosis of sepsis is essential for successful treatment. Hence, we wanted to apply Dynamic Bayesian Networks to the early diagnosis of sepsis at the bedside in the emergency department. We only included the data from the first 24 hours after admission for the test cases, and included only those variables that can be observed at the bedside or can be measured easily in the laboratory within this time. Our goal was to detect the presence or absence of sepsis within 24 hours after admission when the bacterial culture results are often unavailable.

5.1 Overview of Sepsis

The high mortality of sepsis warrants early diagnosis and treatment. Sepsis is responsible for nearly 10% of the ICU admissions in the United States, totaling about 1 million cases nationwide every year[169]. The incidence rate of severe sepsis in the United States is about 300 per 100,000 persons per year, with a total of 750,000 cases nationwide per year. Incidence of sepsis has been estimated to be between 83 and 114 per 100,000 hospital

admissions in Spain, with about 50% mortality rate. In Germany, the incidence of sepsis is about 114 per 100,000 cases per year[170]. These incidence and mortality rates indicate that sepsis is a problem around the world with high mortality and morbidity across various populations. Direct costs per sepsis patient for ICU treatment in the United States have been estimated at more than $40,000. Gram negative bacteria have been implicated as the most common cause, followed by other bacteria and other pathogens.

5.1.1 Diagnosis of Sepsis and Its Complications

The following definitions are from the American College of Chest Physicians (ACCP), and Society of Critical Care Medicine (SCCM) Consensus Conference held in 1991 to define common definitions for sepsis and related disorders and published in 1992[171], here onwards referred to as 'ACCP & SCCM 1992 definitions'. Sepsis is defined as a systemic inflammatory reaction in response to an infection. In addition to SIRS, infection must be present or suspected to confirm a diagnosis of sepsis[171]. SIRS alone is not sufficient to confirm a diagnosis of sepsis, since SIRS can be caused due to noninfectious causes such as pulmonary embolism, adrenal insufficiency, anaphylaxis, pancreatitis, trauma, etc.[171]

In adults, Systemic Inflammatory Response Syndrome (SIRS) is defined as the presence of two or more of the following[171]:

1. Body temperature below 36 C (degrees Celsius) or above 38 C
2. Tachycardia, with heart rate above 90 beats per minute
3. Tachypea (increased respiratory rate), with respiratory rate above 20 per minute, or arterial partial pressure of carbon dioxide ($PaCO_2$) less than 4.3 kPa (kilo Pascals), equivalent to 32 mmHg (millimeters of mercury).
4. White blood cell (WBC) count less than 4,000/mm^3(cubic millimeter) or above 12,000/mm^3, or the presence of more than 10% immature neutrophils (band forms).

When sepsis causes Multiple Organ Dysfunction Syndrome (MODS), such as damage to vital organs, decreased perfusion, or hypotension, it is termed severe sepsis. Sepsis-induced hypotension is defined as a systolic pressure below 90 mmHg or a reduction in

the baseline systolic blood pressure of more than 40 mmHg, in the absence of other causes of hypotension[171].

Sepsis can lead to a condition known as septic shock, which is indicated by hypotension (fall in blood pressure) that is not responsive to fluid replacement or vasopressor drugs[171].

5.1.2 Early Detection of Sepsis in the Emergency Department

At LDS Hospital (LDSH), and Intermountain Medical Center (IMC), two tertiary care hospitals of Intermountain Healthcare in Salt Lake City, Utah, USA, the prevalence of sepsis in patients who directly present at the emergency department is between 1.7% to 2%. Clinical literature shows that patients with sepsis will have high mortality and morbidity if they are not treated immediately and aggressively. However, a confirmatory laboratory test for infections may take several hours to arrive, since culture and sensitivity tests cannot be performed immediately.

Many patients have atypical presentations, and may not have a clear picture of SIRS. To assist the clinicians in detection of sepsis, a clinical decision support system for early detection of sepsis is highly desirable. Sepsis presents a very good case for early detection using clinical decision support systems since the components of SIRS are easily measured at the bedside, or in the case of WBC and band counts, can be obtained in a short amount of time from the laboratory.

We wanted to use a temporal probabilistic model for the early detection of sepsis, and try to understand how the accuracy and certainty of inferences change over time as more data becomes available. We used Dynamic Bayesian Network techniques using the Projeny Toolkit, and our data preparation methods (described in Chapter 3) to create, train and test DBN models for the early detection of sepsis in the emergency department.

5.2 DBN Approach for Sepsis Detection

Sepsis is a rapidly worsening clinical condition. Given the fast rate of change in the physiological parameters, the change in the clinical condition of sepsis patients lends itself well to a temporal probabilistic model such as a Dynamic Bayesian Network. Our

objective was an early detection of sepsis even before many laboratory tests become available, ideally within the first few hours after admission.

5.2.1 Sepsis Data Set

We obtained a data set of about 3,100 patients treated at Intermountain Healthcare, consisting of 20% cases (patients who had sepsis), and 80% controls (patients without sepsis), from Dr. Jason Jones at Intermountain Healthcare. We used the anonymized data set for our sepsis detection modeling. The data elements available in the raw data set were the patients' vital signs (heart rate, respiratory rate, body temperature, systolic blood pressure, diastolic blood pressure, and $PaCO_2$); the patients' lab test results (WBC count, bands percentage); and general encounter information (patient's age, date of admission, date of discharge, etc.). The data set also contained a variable named 'Sepsis', which was entered by a clinician during a retrospective review done for clinical research. Mean blood pressure or mean arterial pressure is the weighted average of the systolic and diastolic pressure. If the mean blood pressure for a specific timeslice for a specific patient was not available, it was calculated using the formula

$$MAP = DP + \frac{SP - DP}{3} \tag{5.1}$$

where MAP denotes mean arterial pressure, SP denotes systolic pressure, and DP represents diastolic pressure. This formula is applicable to adult patients when the blood pressure is not extremely high or low.

The data set we received did not have information about blood (or other specimen) culture and sensitivity results from the microbiology laboratory, information signifying multiorgan dysfunction syndrome (MODS), or treatment information such as the administration of IV fluids and vasopressors, which help with a diagnosis of septic shock. Hence, we did not have the necessary clinical variables for diagnosing severe sepsis or septic shock.

We did not have clinical information denoting suspected or confirmed infection (culture and sensitivity results, clinical notes, etc.). However, we wanted to model a sepsis detection system with the currently available data.

Vital signs were the most numerous type of data in our data set, and they were often measured between 1 to 2 intervals, even though some measurements were up to 20 hours apart. Hence, we used 1 hour as the width of the timeslices in our DBN models to help with early detection of sepsis within a few hours after admission. The lab tests were not measured at such frequent intervals. Not all vital signs were measured at 1-hour intervals. Hence, we had a large amount of missing data in our temporally aggregated and transformed data set.

We applied our data preparation and temporal reasoning techniques described in Chapter 3 to build, train, and test sepsis detection models. We created multiple prepared data sets and temporal models, which showed varying levels of accuracy. Three of these models and prepared data sets provided us the most insight and new lessons. These 3 models, the associated data preparation techniques, the results, and the lessons learned are described in the next 3 sections of this chapter.

5.2.2 Evaluation Technique

Sepsis was a binary variable in our model, with values of true and false. This variable was entered by a clinician, and was considered as the reference standard. The goal of our DBN models was to correctly estimate whether a patient had sepsis or not. Hence, the clinician-entered sepsis diagnosis provided a reference standard against which the DBN models' inferences will be compared. This makes our models similar to laboratory tests that are performed to detect specific diseases.

Given the similarity of our sepsis detection models to laboratory tests that detect a disease, our DBN models can be evaluated using the same evaluation techniques applied to laboratory tests. We decided to perform statistical analysis of our models' inferences in terms of sensitivity, specificity, positive predictive value, negative predictive value, F-value, and area under the ROC (receiver operator characteristic) curve.

The clinician-entered values of sepsis, also considered as the reference standard or 'disease', was left intact in the training data set. The clinician-entered values of sepsis in the test data set were hidden from the inference algorithm during the test iteration, and were later used to validate the inferred values. All other variables were left intact in both the training and test data sets. The state of sepsis estimated by the DBN model was

considered analogous to the lab test finding. If the probability of sepsis estimated by the DBN model was equal to or above 0.5, it was considered as a positive test. If the estimated probability of sepsis was below 0.5, it was considered a negative test.

A 2x2 confusion matrix and standard epidemiologic techniques can then be applied to calculate the sensitivity, specificity, positive predictive value, negative predictive value, and the F-measure. The ROC curve was constructed and the area under the ROC curve was calculated using a procedure described by Morrison using Microsoft Excel[172].

5.3 Experiment 1: Model 1 Using k-means Clustering

The first experiment describes a model created using the clinical description of systemic inflammatory response syndrome (SIRS). The data set was discretized using the k-means clustering technique. The model performed with a fair amount of accuracy for sepsis detection and it revealed various shortcomings that are described, and addressed in subsequent experiments.

5.3.1 Materials and Methods

The structure of the model was designed based on the ACCP & SCCM 1992 definitions of systemic inflammatory response syndrome (SIRS). The evidence of infection was left out of the model. Four binary intermediate nodes were constructed, which denote the four criteria for SIRS. We used the rules defined by ACCP & SCCM 1992 definitions to set the value of these binary nodes for both the training and the test data sets.

The continuous variables in the temporally aggregated and consolidated data were heart rate (beats per minute), body temperature (degrees Celsius), respiratory rate (breaths per minute), $PaCO_2$ (in mmHg), WBC count (per mm^3), and bands (percentage). These continuous variables were discretized using k-means clustering with $k = 10$, and $k = 15$. If there was a significant reduction in error rate for a variable while using 15 bins compared to 10 bins, then we used 15 bins for the variable. Otherwise, we used 10 bins for the variable. It must be noted that the k-means clustering algorithm found less bins than the value of k we provided in some cases. In these cases, the number of bins computed by the algorithm were used for these variables. The model structure is shown in Figure 5.1. The variables HRScore, TachypneaScore, BodyTempScore, WBCScore, and SIRSScore

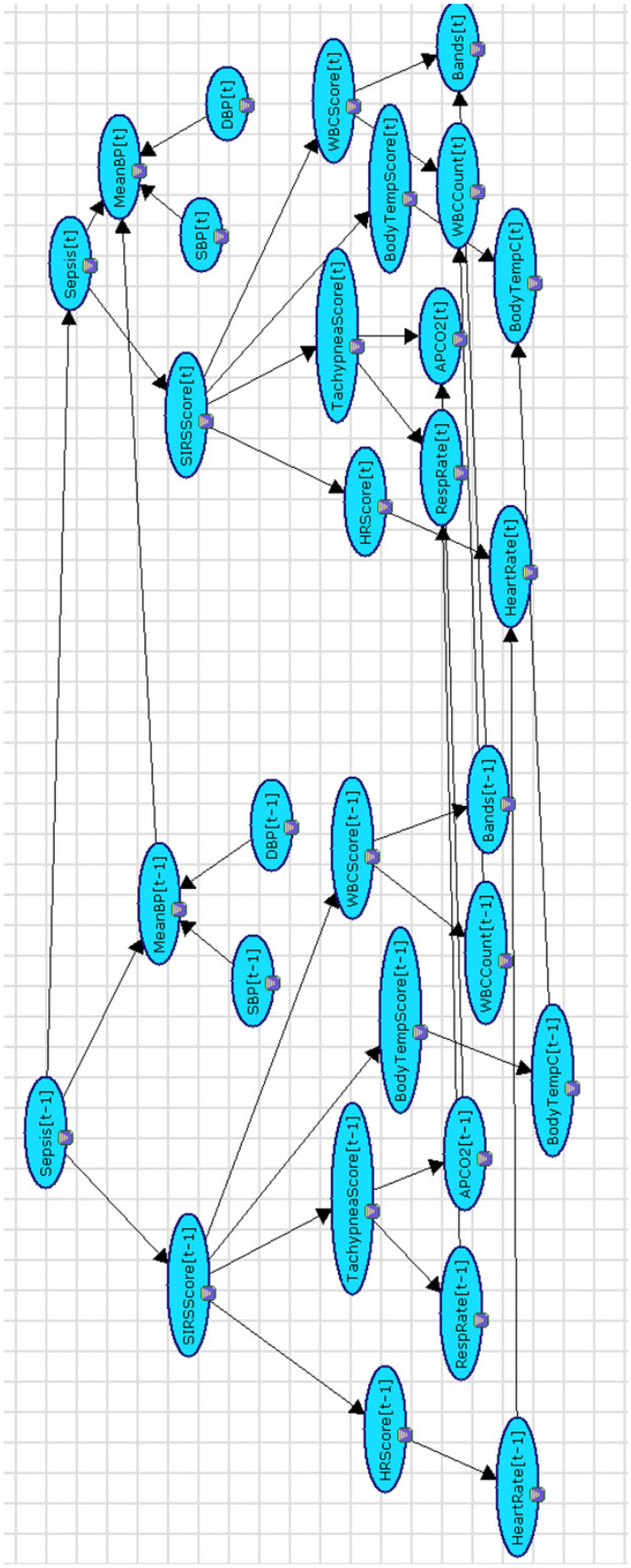


Figure 5.1: Sepsis model 1. SIRS definition, using k-means clustering

were defined as binary variables in accordance with the ACCP & SCCM 1992 definitions, and their values were assigned to true or false in the training data set following the rules specified by these definitions.

The discretized data set was divided into a training data set and a test data set. Two-thirds of anonymized patients were allocated to the training data set at random, and the remaining one-third of the patients were allocated to the test data set. The value of sepsis was replaced with null values for all the patients and all the timeslices in the test data set. The test data set was then divided into four separate data sets having up to 3, up to 6, up to 12, and up to 24 timeslices for each patient. Our goal was to repeat testing with data sets having a different number of timeslices, so that we can simulate testing after the patient has been in the hospital for increasing durations of time (2 hours, 5 hours, 11 hours, and 23 hours, respectively, since the first timeslice was measured when the patient arrived at the emergency department at time = 0 hours after admission).

Training was performed using an EM-based learning algorithm. Training was extremely resource intensive. Training took more than 90 hours, on a computer with two quad-core 2.25GHz (gigahertz) Intel Xeon processors, 24GB (gigabytes) of RAM (Random Access Memory), and 32GB of swap space. Training (parameter learning) ran out of memory with Matlab consuming more than 40GB of memory (RAM and swap space combined, with the rest of the computer's memory taken up by the MySQL database server, Projeny, the desktop environment and other background processes). Hence, the maximum number of timeslices per patient was iteratively pruned in the training data set until the training algorithms executed successfully without running out of memory. We tried pruning to 120 hours, 96 hours, 72 hours, and 48 hours, and the model ran out of memory in all these cases. We finally pruned the training data set to a maximum of 24 timeslices (approximately 1 day). Training completed successfully without out-of-memory errors with this modification.

Testing was performed using a junction tree algorithm. Testing was performed separately with test data sets having a maximum of 3, 6, 12, and 24 timeslices. The probability of sepsis was estimated using the marginal probability distribution calculated by the inference algorithm. Expected values were not calculated (in contrast with the hyperglycemia models described in Chapter 4), since sepsis was a binary, nominal variable. The proba-

bility of sepsis was given by the probability for the sepsis variable to be in the ‘true’ (or ‘sepsis present’) state. If this probability was equal to or above 0.5, it was considered a ‘positive test’. Otherwise, it was considered a ‘negative test’. Statistical analyses were performed as described under ‘Results’.

5.3.2 Results

The model detected sepsis with a fair amount of accuracy. A confusion matrix was plotted with the actual values of sepsis considered as the ‘disease’, and the estimated values of sepsis considered as the ‘test’. The confusion matrix along with the sensitivity, specificity, positive predictive value, negative predictive value, F-measure, and the area under the ROC curve from testing with 3, 6, 12, and 24 timeslices are given in Table 5.1.

The area under the ROC curves for testing model 1 with 3, 6, 12, and 24 timeslices are shown in Figure 5.2.

From the confusion matrices and the ROC curves, we can see that the accuracy of the model increases with availability of more data with the passage of time. The increasing accuracy with time can be easily explained by the fact that we can recognize trends in various physiological parameters and detect a disease with more certainty as time passes. The trends in these parameters with the passage of time are illustrated in Figure 5.3, and their values shown in Table 5.2.

The confusion matrices and ROC curves prove that the model works as expected. However, our goal is early detection of sepsis. We aim to estimate the presence of sepsis with high accuracy within 2 hours of admission (a 3-timeslice model), which is the shortest duration of time supported by our DBN modeling technique. The current model shows a sensitivity of 0.51, a specificity of 0.87, positive predictive value of 0.65, negative predictive value of 0.8 and an area under the curve of 0.63 while testing with 3 timeslices. These measures show that the model is more specific than it is sensitive.

Testing this model with 3 timeslices shows inadequate accuracy. The model also proved to be computationally very expensive, which necessitated the reduction of our training data set. Hence, we modified the model to improve the accuracy and reduce the computational complexity. Two more models were created to achieve these objectives, as shown in the following Sections 5.4, and 5.5.

Table 5.1: Sepsis model 1. Confusion matrices

(a) 3 timeslices

Sepsis	Actual Yes	Actual No	Total
DBN Yes	83	45	128
DBN No	78	308	386
Total	161	353	514
Sensitivity (recall)		83/161	0.5155
Specificity		308/353	0.8725
PPV (precision)		83/128	0.6484
NPV		308/386	0.7979
F-measure			0.5744
AUC			0.6262

(b) 6 timeslices

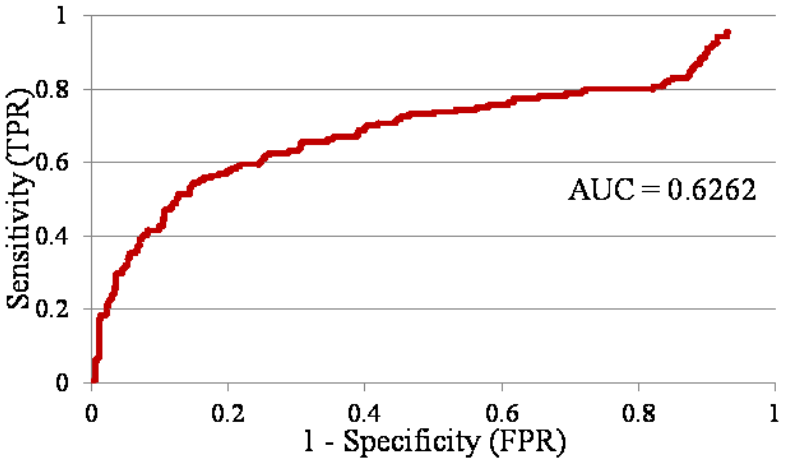
Sepsis	Actual Yes	Actual No	Total
DBN Yes	89	60	149
DBN No	72	293	365
Total	161	353	514
Sensitivity (recall)		89/161	0.5528
Specificity		293/353	0.8300
PPV (precision)		89/149	0.5973
NPV		293/365	0.8027
F-measure			0.5742
AUC			0.6063

(c) 12 timeslices

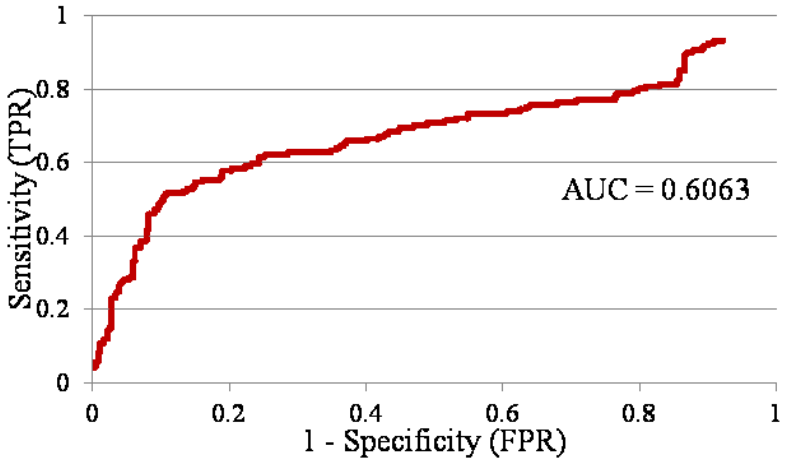
Sepsis	Actual Yes	Actual No	Total
DBN Yes	102	65	167
DBN No	59	288	347
Total	161	353	514
Sensitivity (recall)		102/161	0.6335
Specificity		288/353	0.8159
PPV (precision)		102/167	0.6108
NPV		288/347	0.8300
F-measure			0.6220
AUC			0.7040

(d) 24 timeslices

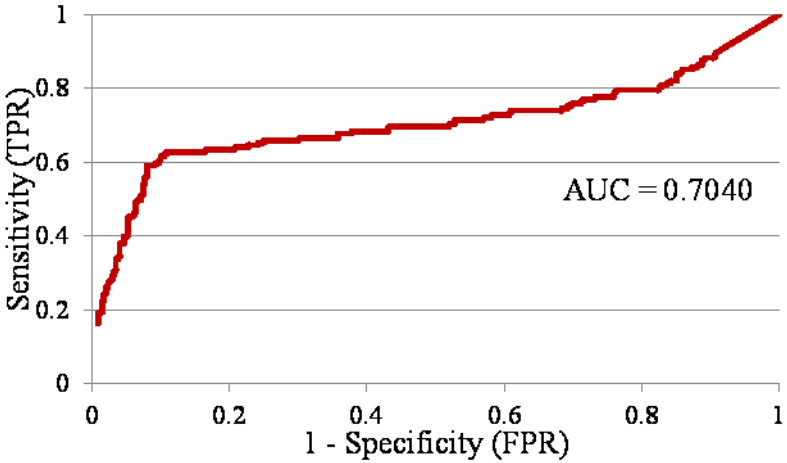
Sepsis	Actual Yes	Actual No	Total
DBN Yes	112	62	174
DBN No	49	291	340
Total	161	353	514
Sensitivity (recall)		112/161	0.6957
Specificity		291/353	0.8244
PPV (precision)		112/174	0.6437
NPV		291/340	0.8559
F-measure			0.6687
AUC			0.7204



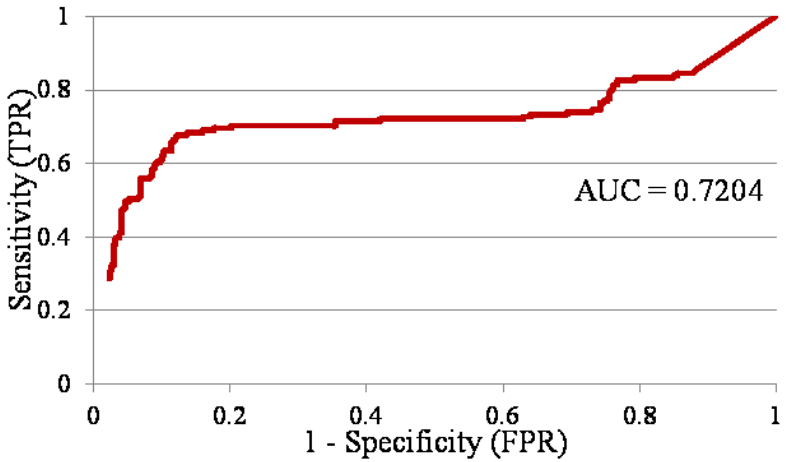
(a) 3 timeslices



(b) 6 timeslices



(c) 12 timeslices



(d) 24 timeslices

Figure 5.2: Sepsis model 1. ROC curves

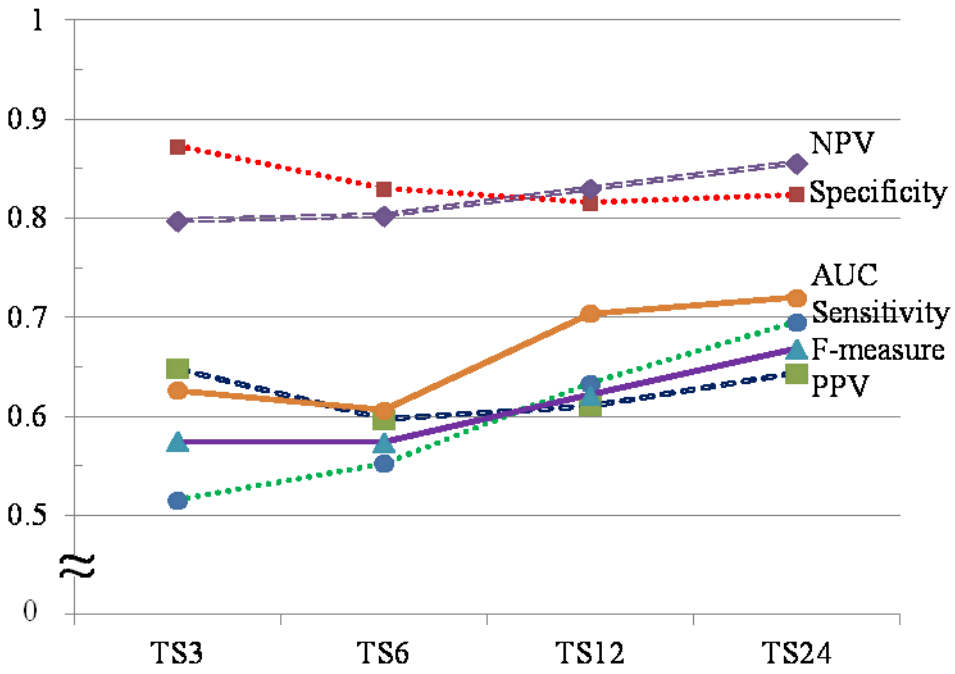


Figure 5.3: Sepsis model 1. Plot of statistical measures over time

Table 5.2: Sepsis model 1. Values of statistical measures over time

	Number of timeslices in test data set			
	TS3	TS6	TS12	TS24
Sensitivity	0.51553	0.55280	0.63354	0.69565
Specificity	0.87252	0.83003	0.81586	0.82436
PPV (precision)	0.64844	0.59732	0.61078	0.64368
NPV	0.79793	0.80274	0.82997	0.85588
F-measure	0.57439	0.57419	0.62195	0.66866
AUC	0.62620	0.60635	0.70398	0.72041

5.4 Experiment 2: Model 2 Using MDL Discretization

A large number of nodes and states, along with the intermediate nodes and missing data, made model 1 computationally very expensive for parameter learning. Transforming all the physiological parameters from continuous variables into binary intermediate nodes led to loss of information, which in turn lowered the accuracy of the model. The missing data problem cannot be overcome while using the same data set. However, the discretization technique and the model structure can be changed to make the model computationally more tractable and more accurate. We made these changes in model 2, and performed this

experiment.

5.4.1 Materials and Methods

The ACCP & SCCM 1992 definitions of systemic inflammatory response syndrome (SIRS) are intended for human use at the bedside. The simple nature of the rules, transforming continuous variables into binary variables, and then satisfying the constraint of 'any two out of four' conditions is intuitive and user-friendly for the human experts to compute in their minds. However, transforming continuous variables into binary intermediate variables led to the loss of large amounts of information, which lowered the accuracy of the model.

K-means clustering tries to find natural clustering patterns in the input data, and fits the input data set into a finite set of symbols (or discrete states). This process is helpful to reduce the dimensionality of the model. However, the k-means clustering process does not consider how the variation in an independent variable (e.g., body temperature) affects the variation in a dependent variable (e.g., sepsis). The variation in the dependent variable may be explained by a smaller number of states than the number of states (up to 10 or 15) used for k-means clustering. Hence, we looked for algorithms that discretize the continuous independent variables in context of variation of the dependent variable.

We found that the Minimum Description Length (MDL) algorithm described by Fayyad and Irani[152] divides the independent variable into a minimal number of states required to explain the variation in the dependent variable. The MDL algorithm produced a much smaller number of states for all the continuous variables. A smaller state-space reduces the computational complexity of parameter learning.

The continuous variables in the temporally aggregated and consolidated data were heart rate (beats per minute), body temperature (degrees Celsius), respiratory rate (breaths per minute), $PaCO_2$ (mmHg), WBC count (mm^3), and bands (percentage). These were discretized using Fayyad and Irani's MDL algorithm implemented in Weka. From clinical literature, it is observed that the patient's age affects how the patient's body responds to sepsis. A child, a young adult, and an elderly person will respond in different ways to sepsis, and the patterns seen in their heart rate, blood pressure, temperature, WBC count, etc. will be different even if they have similar sepsis conditions. Hence, we included

the patient's age as one of the variables in the model. We connected patient's age to the various nodes representing various physiological parameters, and to sepsis itself. The model structure is shown in Figure 5.4.

Training and testing were done in the same way as experiment 1. The discretized data set was divided into a training data set and a test data set. Two-thirds of anonymized patients were allocated to the training data set at random, and the remaining one-third of the patients were allocated to the test data set at random. The value of sepsis was replaced with null values for all the patients and all the timeslices in the test data set. The test data set was then divided into four separate data sets having up to 3, up to 6, up to 12, and up to 24 timeslices for each patient. Our goal was to repeat testing with data sets having a different number of timeslices, so that we can simulate testing after the patient has been in the hospital for increasing durations of time (2 hours, 5 hours, 11 hours, and 23 hours, respectively, since the first timeslice was measured when the patient arrived at the emergency department at time = 0 hours after admission).

Training was performed using an EM-based learning algorithm. Training was initially performed with a data set having a maximum of 24 timeslices for each patient. Training was much less resource intensive compared to model 1. Training took about 8 hours, on the same computer with two quad-core 2.25GHz Intel Xeon processors, 24GB of RAM, and 32GB of swap space. Training (parameter learning) required much less memory with Matlab consuming less than 7GB of memory. Hence, the maximum number of timeslices per patient was increased in the training data set to the finally chosen value of 72 timeslices (approximately 3 days). Training completed successfully in less than 10 hours, with less than 7GB of memory utilization with this modification.

Testing was performed using a junction-tree algorithm. Testing was performed as in model 1 with test data sets having a maximum of 3, 6, 12, and 24 timeslices, since our goal was to predict sepsis within 24 hours after admission. The probability of sepsis was estimated using the marginal probability distribution calculated by the inference algorithm. Expected values were not calculated (in contrast with the hyperglycemia models described in Chapter 4), since sepsis was a binary, nominal variable. The probability of sepsis was given by the probability for the sepsis variable to be in the 'true' (or 'sepsis present') state. If this probability was equal to or above 0.5, it was considered a 'positive test'. Otherwise,

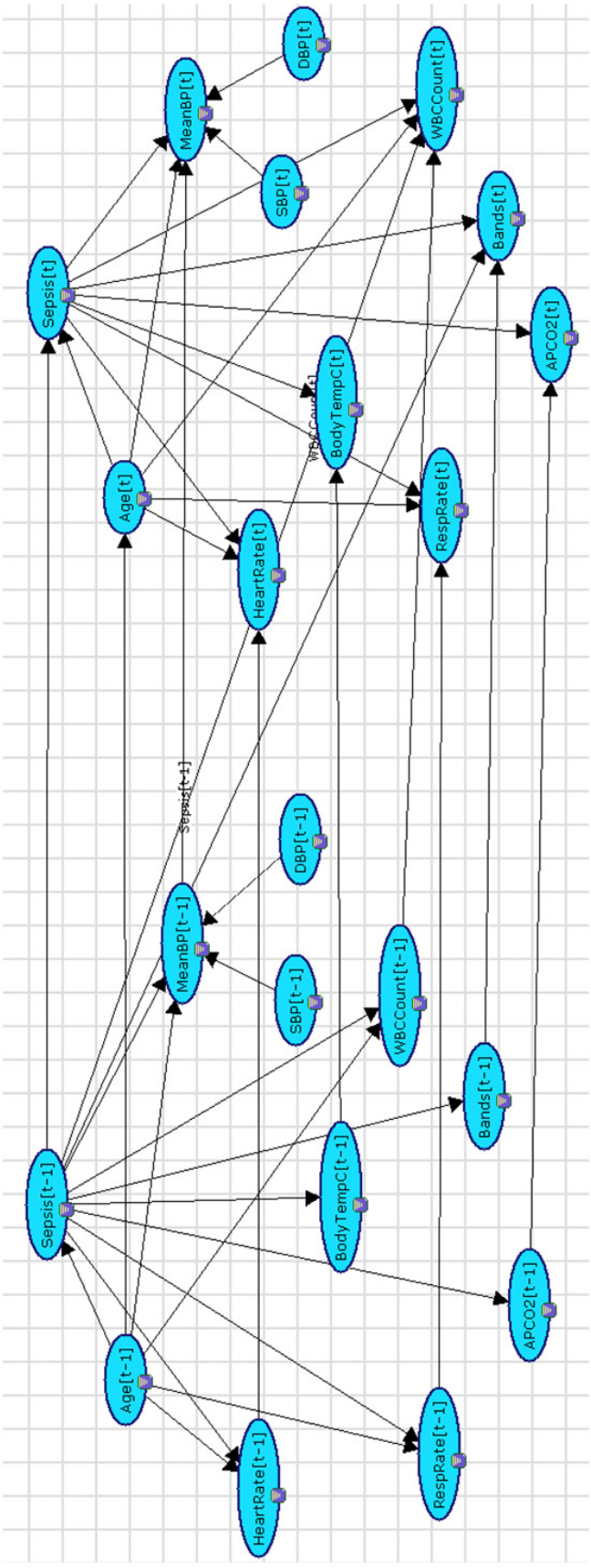


Figure 5.4: Sepsis model 2. SIRS with age, using MDL discretization

it was considered a ‘negative test’. Statistical analyses were performed as described under ‘Results’.

5.4.2 Results

The model estimated sepsis with significantly higher accuracy than model 1. A confusion matrix was plotted with the actual values of sepsis considered as the ‘disease’, and the estimated values of sepsis considered as the ‘test’. The confusion matrix along with the sensitivity, specificity, positive predictive value, negative predictive value, F-measure, and the area under the ROC curve from testing with 3, 6, 12, and 24 timeslices are given in Table 5.3. The area under the ROC curves for testing model 2 with 3, 6, 12 and 24 timeslices are shown in Figure 5.5.

From the confusion matrices and the ROC curves, we can see that the accuracy of prediction of model 2 is significantly better than that of model 1. We can also see that the accuracy increases with availability of more data with the passage of time. The trends in these parameters with the passage of time are illustrated in Figure 5.6, and their values shown in Table 5.4.

A comparison of the accuracy of sepsis detection of the current model (model 2) with that of the previous model (model 1) shows that using the MDL algorithm for data discretization and removing the binary intermediate nodes from the model structure helped to reduce the loss of information and to significantly improve the accuracy of the model. We also see that reducing the size of the state-space significantly reduced the computational complexity of parameter learning.

Our goal, as in the previous model, is early detection of sepsis. We aim to detect sepsis with high accuracy within 2 hours of admission (a 3-timeslice model), which is the shortest duration of time supported by our DBN modeling technique. The current model shows a sensitivity of 0.6, a specificity of 0.94, positive predictive value of 0.65, negative predictive value of 0.93, and an area under the curve of 0.81 while testing with 3 timeslices. All the parameters except the sensitivity improved compared to model 1. These measures again show that the model is more specific than it is sensitive.

However, we found some problems with the structure of the model that can lead to a reduction in accuracy. A detailed discussion of ‘explaining away’, and d-separation is

Table 5.3: Sepsis model 2. Confusion matrices

(a) 3 timeslices

Sepsis	Actual Yes	Actual No	Total
DBN Yes	99	54	153
DBN No	65	825	890
Total	164	879	1043
Sensitivity (recall)		99/164	0.6037
Specificity		825/879	0.9386
PPV (precision)		99/153	0.6471
NPV		825/890	0.9270
F-measure			0.6246
AUC			0.8126

(b) 6 timeslices

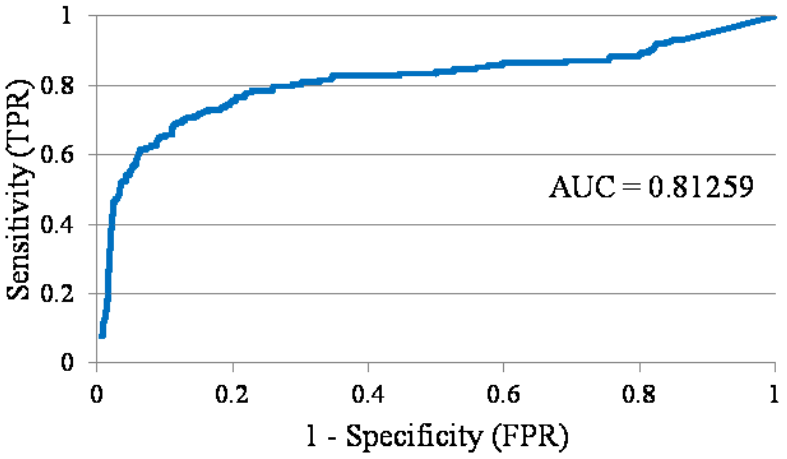
Sepsis	Actual Yes	Actual No	Total
DBN Yes	105	57	162
DBN No	59	822	881
Total	164	879	1043
Sensitivity (recall)		105/164	0.6402
Specificity		822/879	0.9352
PPV (precision)		105/162	0.6481
NPV		822/881	0.9330
F-measure			0.6442
AUC			0.8238

(c) 12 timeslices

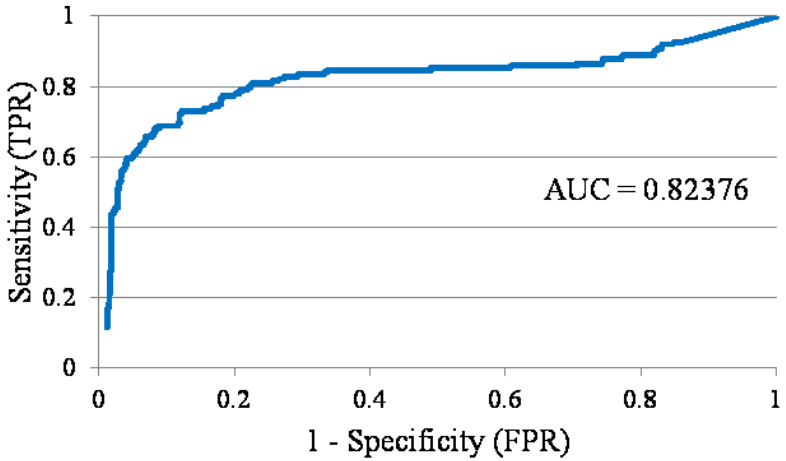
Sepsis	Actual Yes	Actual No	Total
DBN Yes	117	61	178
DBN No	47	818	865
Total	164	879	1043
Sensitivity (recall)		117/164	0.7134
Specificity		818/879	0.9306
PPV (precision)		117/178	0.6573
NPV		818/865	0.9457
F-measure			0.6842
AUC			0.8349

(d) 24 timeslices

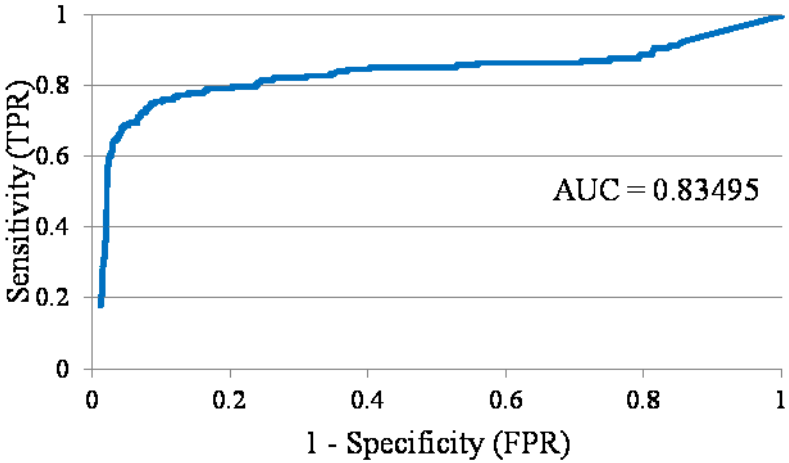
Sepsis	Actual Yes	Actual No	Total
DBN Yes	127	62	189
DBN No	37	817	854
Total	164	879	1043
Sensitivity (recall)		127/164	0.7744
Specificity		817/879	0.9295
PPV (precision)		127/189	0.6720
NPV		817/854	0.9567
F-measure			0.7195
AUC			0.8430



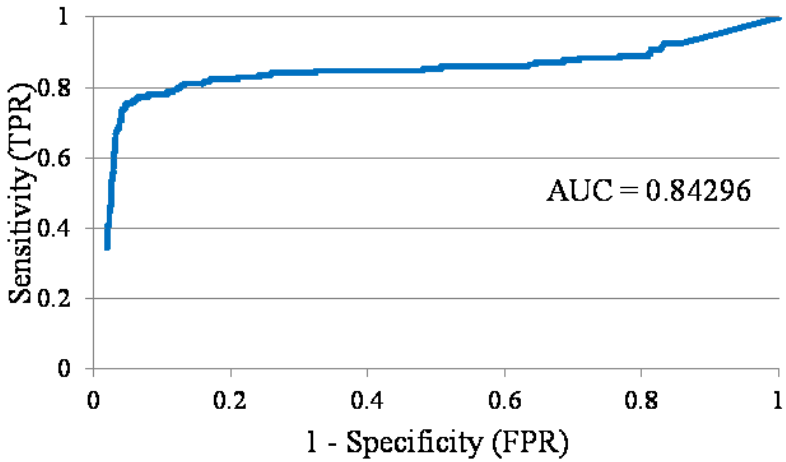
(a) 3 timeslices



(b) 6 timeslices



(c) 12 timeslices



(d) 24 timeslices

Figure 5.5: Sepsis model 2. ROC curves

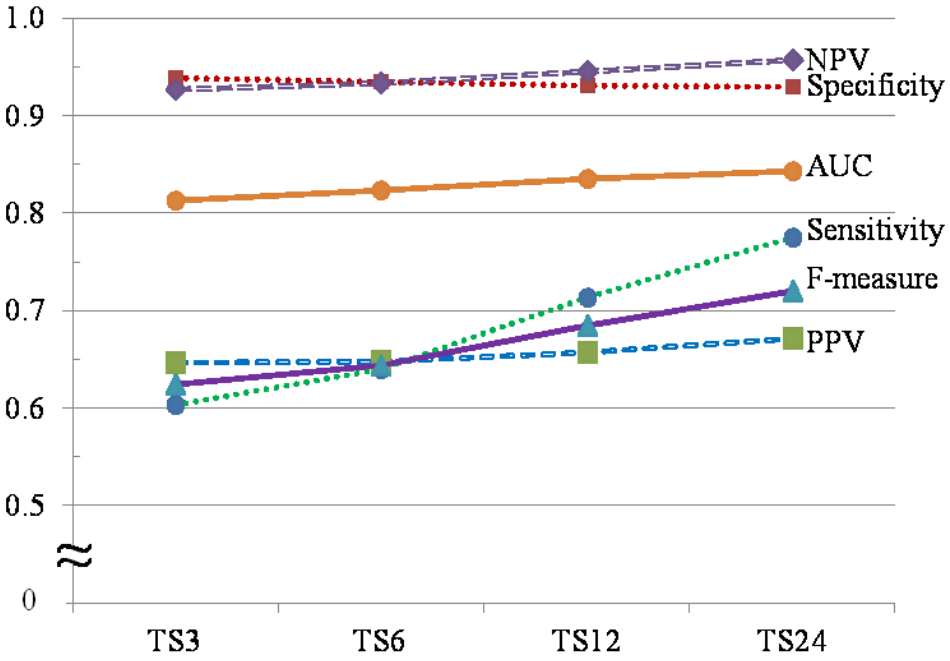


Figure 5.6: Sepsis model 2. Plot of statistical measures over time

Table 5.4: Sepsis model 2. Values of statistical measures over time

	Number of timeslices in test data set			
	TS3	TS6	TS12	TS24
Sensitivity	0.60366	0.64024	0.71341	0.77439
Specificity	0.93857	0.93515	0.93060	0.92947
PPV (precision)	0.64706	0.64815	0.65730	0.67196
NPV	0.92697	0.93303	0.94566	0.95667
F-measure	0.62461	0.64417	0.68421	0.71955
AUC	0.81259	0.82376	0.83495	0.84295

presented in Chapter 2. Mean blood pressure was explained away by systolic and diastolic pressure due to the way the mean blood pressure was related to sepsis, systolic pressure, and diastolic pressure. Explanation of mean blood pressure by sepsis is hence weakened.

We had also made age a parent of sepsis in this model. We found that connecting age directly to sepsis will make age explain away some of the variation in sepsis. Our goal was to explain the variation in physiological parameters due to sepsis, and how this interaction is influenced by age. Age does not by itself cause sepsis, and hence the edge from age to sepsis leads to learning spurious conditional probabilities.

We fixed the above structural issues in the model, and repeated the training and testing, as described in experiment 3.

5.5 Experiment 3: Model 3 Using MDL Discretization

Model 2 was more accurate in detecting sepsis than model 1 with just 3 timeslices (2 hours) of patient data. The model may be very useful in detecting sepsis in the ICU. However, issues with conditional independence (d-separation, explaining away) reduced the accuracy of the model. Hence, we wanted to correct the model structure and repeat the experiment. These changes led us to model 3, and the training, testing, and results are described as experiment 3.

5.5.1 Materials and Methods

In our model, mean blood pressure is a variable that abstracts systolic pressure and diastolic pressure, and their relationship to sepsis. The direction of the relationships caused an incorrect representation of conditional independence. Therefore, we removed mean arterial pressure, which is fully explained by systolic and diastolic blood pressure, from model 3. We then created edges from sepsis to systolic pressure and diastolic pressure. The edge from age to sepsis indicates a spurious conditional dependence. Hence, we removed the edge between age and sepsis. In model 3, age and sepsis are d-separated by the nodes that represent systolic blood pressure, diastolic blood pressure, heart rate, respiratory rate, and WBC count. Age and sepsis together explain the variation in these physiological parameters in the model. If none of these five physiological parameters in the model are known, then age and sepsis are mathematically conditionally independent, as described in Section 2.4.2.

We used the same data set used in experiment 2 to perform experiment 3. The continuous variables in the temporally aggregated and consolidated data were heart rate (beats per minute), body temperature (degrees Celsius), respiratory rate (breaths per minute), $PaCO_2$ (in mmHg), WBC count (per mm^3), and bands (percentage). These were discretized using Fayyad and Irani's MDL algorithm implemented in Weka. The model structure is shown in Figure 5.7.

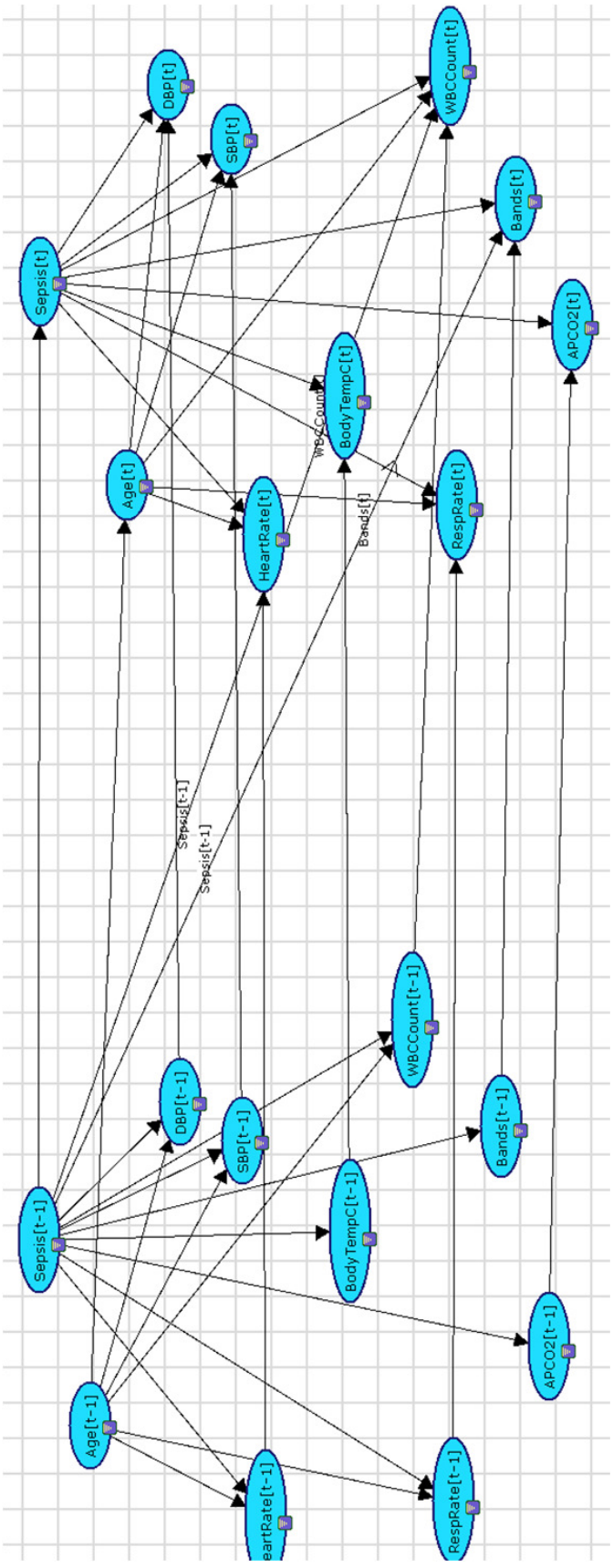


Figure 5.7: Sepsis model 3. Modified SIRS with age, using MDL discretization

Training and testing were done in the same way as experiment 2, using the same data sets. The discretized data set was divided into training data set and test data set. Two-thirds of anonymized patients were allocated to the training data set at random, and the remaining one-third of the patients were allocated to the test data set. The value of sepsis was replaced with null values for all the patients and all the timeslices in the test data set. The test data set was then divided into four separate data sets having up to 3, up to 6, up to 12, and up to 24 timeslices for each patient. Our goal was to repeat testing with data sets having different number of timeslices, to that we can simulate testing after the patient has been in the hospital for increasing durations of time (2 hours, 5 hours, 11 hours, and 23 hours, respectively, since the first timeslice was measured when the patient arrived at the emergency department at time = 0 hours after admission).

Training was performed using an EM-based learning algorithm. Since model 2 was computationally less expensive than model 1, we increased the number of timeslices in the training data set for model 3. Training was performed with a data set having a maximum of 168 timeslices (approximately 7 days since admission) for each patient. Training was almost as resource intensive as model 2, which is significantly less resource intensive than model 1. Training took about 9 hours, on the same computer with two quad-core 2.25GHz Intel Xeon processors, 24GB of RAM, and 32GB of swap space. Training (parameter learning) required much less memory than model 1 but similar amount of memory as model 2, with Matlab consuming less than 7GB of memory. Training completed successfully in less than 9 hours, with less than 7GB of memory utilization with this modification.

Testing was performed using a junction-tree algorithm. Testing was performed as before in models 1 and 2 with test data sets having a maximum of 3, 6, 12, 24, and 48 timeslices. Our goal was to detect sepsis within 24 hours after admission. However, in contrast to models 1 and 2, we included a test data set with 48 timeslices to study how increased the duration of the test data set affects the results.

The probability of sepsis was estimated using the marginal probability distribution calculated by the inference algorithm. Expected values were not calculated (in contrast with the hyperglycemia models described in Chapter 4), since sepsis was a binary, nominal variable. The probability of sepsis was given by the probability for the sepsis variable to

be in the ‘true’ (or “sepsis present”) state. If this probability was equal to or above 0.5, it was considered a ‘positive test’. Otherwise, it was considered a ‘negative test’. Statistical analyses were performed as described under ‘Results’.

5.5.2 Results

The model detected sepsis with significantly higher accuracy than both model 1 and model 2. A confusion matrix was plotted with the actual values of sepsis considered as the ‘disease’, and the estimated values of sepsis considered as the ‘test’. The confusion matrix along with the sensitivity, specificity, positive predictive value, negative predictive value, F-measure, and the area under the ROC curve from testing with 3, 6, 12 and 24 timeslices are given in Table 5.5. The area under the ROC curves for testing model 2 with 3, 6, 12, and 24 timeslices are shown in Figure 5.8.

From the confusion matrices, and the ROC curves, we can see that the accuracy of sepsis detection of model 3 is significantly better than that of both model 1 and model 2. We can also see that the accuracy increases with availability of more data with the passage of time. The trends in these parameters with the passage of time are illustrated in Figure 5.9, and their values shown in Table 5.6. Table 5.6 also includes the results from the test data set with 48 timeslices.

A comparison of the accuracy of sepsis detection of the current model (model 3) with that of the previous models (models 1 and 2) shows that using the MDL algorithm and a clinically accurate model structure helped to reduce the loss of information and avoid the conditional independence problems, and to significantly improve the accuracy of the model.

Our goal, as in the previous models, is early detection of sepsis. We aim to detect sepsis with high accuracy within 2 hours of admission (a 3-timeslice model), which is the shortest duration of time supported by our DBN modeling technique. The current model shows a sensitivity of 0.69, a specificity of 0.95, positive predictive value of 0.72, negative predictive value of 0.94, and an area under the curve of 0.91 while testing with 3 timeslices. All the parameters including the sensitivity improved compared to model 2. These measures again show that the model is more specific than it is sensitive. However, it shows much higher sensitivity than models 1 and 2.

Table 5.5: Sepsis model 3. Confusion matrices

(a) 3 timeslices

Sepsis	Actual Yes	Actual No	Total
DBN Yes	113	45	158
DBN No	51	834	885
Total	164	879	1043
Sensitivity (recall)		113/164	0.6890
Specificity		834/879	0.9488
PPV (precision)		113/158	0.7152
NPV		834/885	0.9424
F-measure			0.7019
AUC			0.9110

(b) 6 timeslices

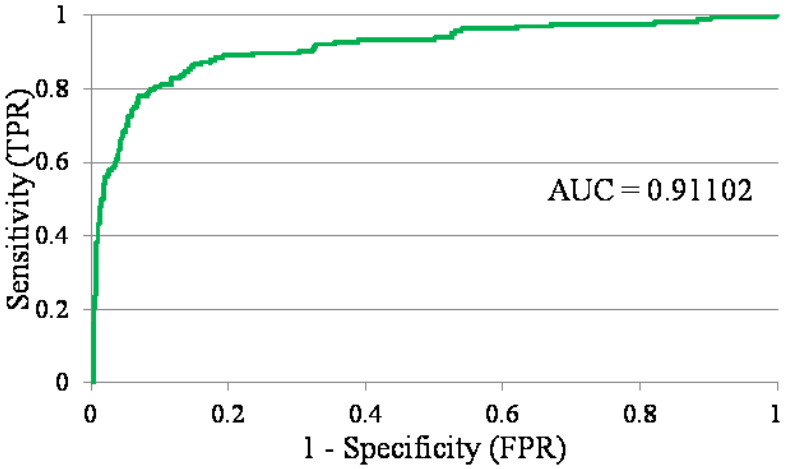
Sepsis	Actual Yes	Actual No	Total
DBN Yes	116	44	160
DBN No	48	835	883
Total	164	879	1043
Sensitivity (recall)		116/164	0.7073
Specificity		835/879	0.9499
PPV (precision)		116/160	0.7250
NPV		835/883	0.9456
F-measure			0.7160
AUC			0.9150

(c) 12 timeslices

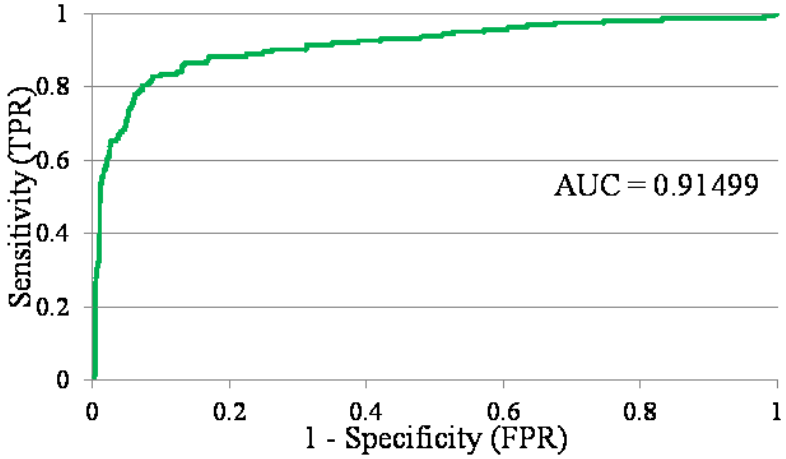
Sepsis	Actual Yes	Actual No	Total
DBN Yes	134	45	179
DBN No	30	834	864
Total	164	879	1043
Sensitivity (recall)		134/164	0.8171
Specificity		834/879	0.9488
PPV (precision)		134/179	0.7486
NPV		834/864	0.9653
F-measure			0.7813
AUC			0.9336

(d) 24 timeslices

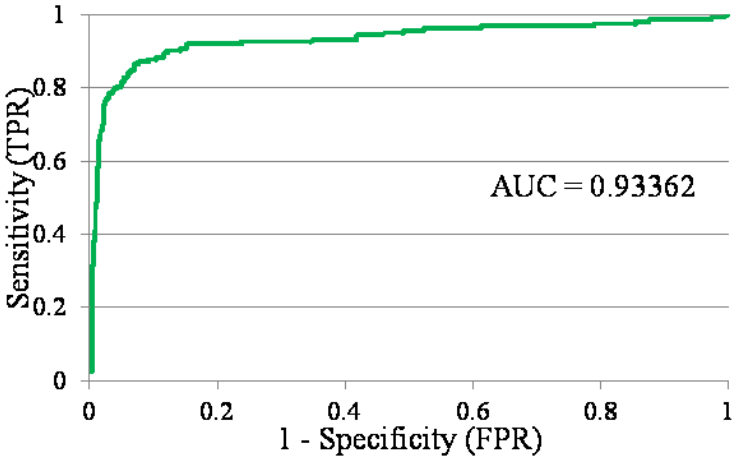
Sepsis	Actual Yes	Actual No	Total
DBN Yes	141	48	189
DBN No	23	831	854
Total	164	879	1043
Sensitivity (recall)		141/164	0.8598
Specificity		831/879	0.9454
PPV (precision)		141/189	0.7460
NPV		831/854	0.9731
F-measure			0.7989
AUC			0.9435



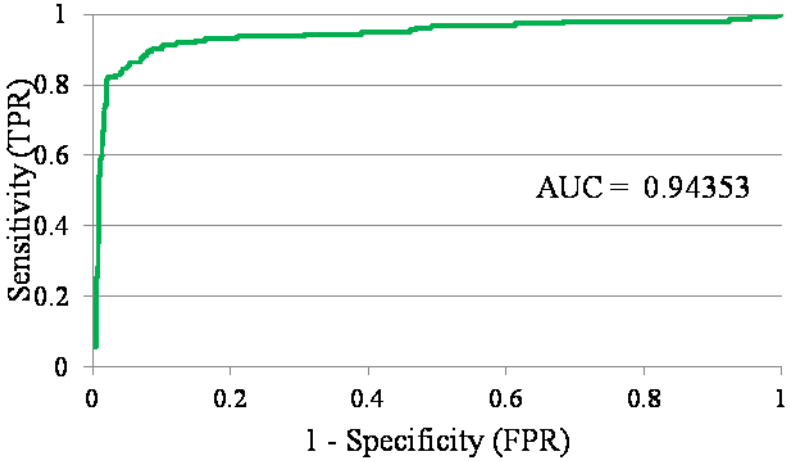
(a) 3 timeslices



(b) 6 timeslices



(c) 12 timeslices



(d) 24 timeslices

Figure 5.8: Sepsis model 3. ROC curves

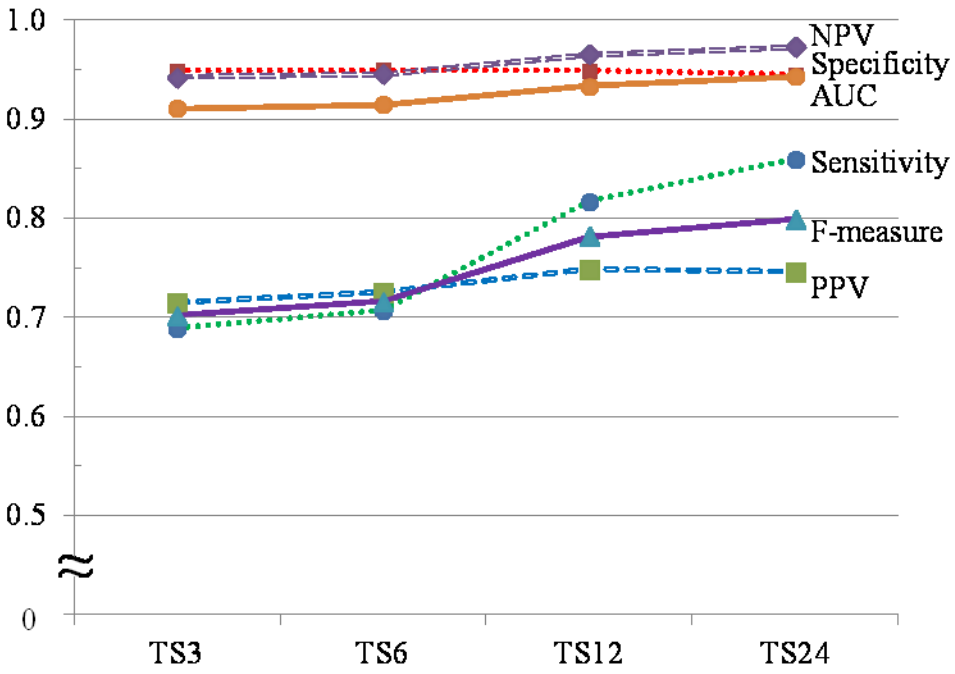


Figure 5.9: Sepsis model 3. Plot of statistical measures over time

Table 5.6: Sepsis model 3. Values of statistical measures over time

	Number of timeslices in test data set				
	TS3	TS6	TS12	TS24	TS48
Sensitivity	0.68902	0.70732	0.81707	0.85976	0.88415
Specificity	0.94881	0.94994	0.94881	0.94539	0.94198
PPV (precision)	0.71519	0.72500	0.74860	0.74603	0.73980
NPV	0.94237	0.94564	0.96528	0.97307	0.97757
F-measure	0.70186	0.71605	0.78134	0.79887	0.80556
AUC	0.91102	0.91499	0.93362	0.94353	0.93029

5.6 Discussion

From the 3 models above, we can demonstrate that DBN methods can be used to successfully detect sepsis in patients in the emergency department within 2 hours of admission. The model can detect sepsis with variables that are mostly collected at the bedside, and WBC count and bands percentage, which are easily obtained from the lab in a short duration of time. Of particular note is the results from testing using 3 timeslices in the third experiment. This test iteration reflects inference using just 2-hours' data from the moment the patient is admitted to the emergency department. This test iteration showed

a sensitivity of 0.69, a specificity of 0.95, positive predictive value of 0.72, negative predictive value of 0.94, and an area under the ROC curve of 0.91. This model may be suitable for use in the real-world clinical setting for early detection of sepsis if it overcomes the following limitation.

It must be noted that the prior probability of sepsis in our data set was 0.2, because our data set had 20% cases and 80% controls. In the emergency department of Intermountain Healthcare, the prevalence of sepsis is about 2%, which translates into a prior probability of 0.02. The real-world prior probability of sepsis is only 1/10 of the prior probability in our data set. Hence, the PPV and NPV of the model will change in the real world. Hence, further testing and validation needs to be done before this model may be used in an emergency department to detect sepsis.

The models and experiments above provide lessons to reduce the computational complexity of DBN learning and inference, and at the same time improve the predictive accuracy. The experiments show that both data preparation and model structure affect the accuracy as well as the computational complexity of the model. The experiments also show that it is possible to reduce the size of the state-space without a corresponding reduction in accuracy. It is also shown that simple models that are designed to be intuitive for human experts such as the SIRS model may not be computationally efficient or accurate for probabilistic modeling. Models that take into account the complex conditional interdependencies and reflect them accurately, while defining the state-space in a meaningful way, prove to be more accurate in probabilistic learning and inference.

We have shown that by using more meaningful methods of data preparation such as the MDL algorithm which analyzes the variation in the dependent variable to discretize the independent variable, we can obtain smaller but more meaningful state-spaces. This is a novel finding in the area of medicine, where probabilistic temporal reasoning methods have not been used extensively. A majority of probabilistic temporal reasoning experiments in the biomedical domain use Hidden Markov Models which in turn use equal interval and equal frequency discretization. Hence, our experiments add new knowledge to the area of probabilistic temporal reasoning in medicine, by proving that it is possible to use more versatile models such as DBNs, with discretization techniques such as the MDL algorithm, to accurately detect clinical conditions.

CHAPTER 6

CONCLUSIONS

Dynamic Bayesian Networks generalize a large class of probabilistic temporal reasoning techniques that include Hidden Markov Models and Kalman Filter Models. DBNs provide a powerful formalism to perform learning and inference with models that have complex probabilistic relationships within and across instances of time. Pathophysiological processes and clinical practice workflow are inherently temporal processes. The complexity of medical science and the practice of medicine prompt the need for clinical decision support tools that can help with temporal modeling and prediction. Dynamic Bayesian Networks are an ideal candidate for application in the medical domain to address these challenges.

In spite of the success of DBNs and related techniques in other fields such as engineering, finance, economics, speech recognition, and genomic and proteomic modeling, they have not been used to a significant extent in clinical medicine. Challenges to their adoption include the difficulty in modeling clinical processes using a temporal model, creating the model structure, data aggregation, consolidation and discretization, support for variable length temporal processes, learning and inference with missing data, and ease of data binding for learning and inference. These are the challenges that motivated this research. This research addresses most of these challenges, as described in this dissertation.

6.1 Novel Contributions of this Research

6.1.1 Model Structure

We learned through our models and experiments that both model structure and data preparation can affect the computational tractability and predictive accuracy of the model. The structure of the model with discrete variables includes the nodes, edges, and the states of the model. The nodes and edges can often be discovered from medical literature or

by interviewing clinicians. We described with evidence the need to avoid conditional independence and d-separation issues, and the problem of 'explaining away', and how to solve these issues.

6.1.2 Temporal Data Aggregation, Consolidation, and Abstraction

Temporal data aggregation and consolidation techniques were also explored in detail. We described data preparation techniques that can be generalized to temporal reasoning problems in medicine. We described methods that can be used to aggregate and consolidate clinical temporal data from many different data sources into a uniform denormalized relational database table. We then described a method to perform temporal abstraction to select a representative data point for each time interval.

6.1.3 Discretization of Continuous Variables

The states of discrete variables are not easy to define. Discretization is a double-edged sword that makes the model amenable to learning and inference by various algorithms, but it leads to loss of information and introduction of noise at the same time. Discretization techniques that are suitable for temporal probabilistic reasoning given the complexity and the sparse nature of clinical data have not been studied previously. Classical techniques such as equal interval and equal frequency discretization do not always yield the best results.

We tested a combination of domain-based and equal interval discretization, and we compared it to k-means clustering. We also separately compared k-means clustering to MDL discretization. We discovered that k-means clustering and MDL discretization techniques prove to be appropriate for use in the medical domain compared to domain-based and equal interval discretization. We also demonstrated that the MDL discretization technique leads to more accurate and more tractable temporal models. These findings may be generalizable to other temporal models in the medical domain, and may be amenable to being performed in an automated way regardless of the nature of the variable.

6.1.4 Computational Tractability

We described the computational tractability of various modeling techniques and discretization techniques. We described how the structure of the model and data preparation can be improved to make the model computationally more tractable. We performed these tasks without a reduction in predictive accuracy of the model. We demonstrated that it is possible to optimize a model to improve both the computational tractability and the accuracy.

6.1.5 Validation Techniques

We described how to process the results and validate them by calculating the expected value and the statistical measures. These processes are performed automatically by several commercial or proprietary toolkits. We described how to perform this task from starting with the raw results, namely the marginal probability distributions. These techniques can be generalized to perform validation from the marginal probability distributions obtained using any other temporal model or toolkit.

6.1.6 Insulin Dosing and Glucose Control in the ICU

We created a model starting with medical literature, and trained and tested it using the above-mentioned techniques in two test cases. The first test case, insulin dosing and glucose control in the ICU for patients with stress-induced hyperglycemia, showed very high accuracy. Our model performed as well as eProtocol-insulin, the existing gold standard at Intermountain Healthcare. Our model provided clinically valid insulin doses as validated by comparing them to the insulin doses recommended by eProtocol-insulin. It is to be noted that our model performed well even without the patients' feeding data, which were available to eProtocol-insulin. This model shows excellent promise for use in a real-world clinical setting. A computational model with comparably high accuracy to predict serum glucose or recommend insulin dose that has been tested with a similarly large corpus of real clinical data have not been currently described in medical literature.

6.1.7 Early Prediction of Sepsis in the ED

Our model predicted sepsis with very high accuracy even with just the first 2 hours' data from the instance when a patient is admitted to the emergency department. The model predicted sepsis with high accuracy even in the absence of culture and sensitivity results. The accuracy of our model increased if the patients' data of a longer duration is available to the inference algorithm, which reflects the certainty of diagnosis made by physicians in clinical practice. We also showed with multiple experiments the methods to improve both the computational tractability and accuracy of the model. A computational model with comparably high accuracy to predict sepsis that has been tested with a similarly large corpus of real clinical data have not been currently described in medical literature.

6.1.8 Projeny, an Open Source Temporal Reasoning Toolkit

Projeny, the probabilistic networks generator in Java, is an open-source toolkit that was created as part of this research. The toolkit is based on three other open-source toolkits. Projeny has been released as open-source, and is regularly updated with new features and bug fixes. Projeny allows the user to easily create the nodes and edges in a DBN model, define the states, and define which nodes are observed and which nodes are hidden. Projeny also allows the user to perform data binding in a way that is easier than most probabilistic modeling tools, whether free or proprietary. Projeny also allows the user to perform data binding for both training and testing, and saves the model in an easily interoperable XML format. The modeling, learning, and inference tasks of all the experiments in this dissertation were performed using Projeny, proving the validity and usefulness of the tool. Projeny has already been downloaded more than 100 times and is being used by other DBN researchers.

6.1.9 Generalizability

Generalizability and external validity were among the main goals of this research. The methods described in this research and tools created during this research are generalizable to other similar temporal reasoning problems. The generalizability of the techniques have partly been proved by applying them to the two test cases described in this dissertation. Further generalizability and external validity for dissimilar problems (e.g., large time

horizons, higher order Markov processes, etc.) remain to be tested.

6.1.10 Nonoriginal Contributions of this Dissertation

Chapter 2 of this dissertation provides a comprehensive summary of temporal reasoning and representation. It serves as a starting point to a researcher who wants to explore temporal representation, temporal logic, or temporal reasoning and prediction.

6.2 Limitations

Several limitations of the methods and tools described in this dissertation have been identified. The following sections discuss these limitations.

6.2.1 Limitations of the Learning and Inference Algorithms

The models and tools only support discrete nodes at present, due to a limitation of the learning and inference algorithms that are used. Continuous nodes are not supported. The learning and inference algorithms used in this experiment, and the Projeny tool, only support first-order Markov processes. Higher order Markov processes are not currently supported. The algorithms also do not support continuous-time DBNs or DBN models with timeslices of variable width.

6.2.2 Limitations of the Projeny Tool

Currently, the tool only supports one exact algorithm each for parameter learning and inference. Approximate learning and inference algorithms, and other exact learning and inference algorithms are not currently implemented. The other limitations of the learning and inference algorithms apply to Projeny due to its dependence on these algorithms.

6.2.3 Utility and Action Nodes

The models and tools described do not support utility and action nodes. By extension, POMDPs and LIMIDs are not supported.

6.3 Future Research

Further research areas include new algorithms, new modeling techniques, implementation of new features in the tool, and new or extended applications.

We hope to support more learning and inference algorithms in Projeny and to apply these algorithms for parameter learning and inference. The computational tractability and accuracy of these algorithms may be tested, and compared to the currently used algorithms.

We intend to perform more detailed comparisons and evaluations of various discretization algorithms using the same model and data sets. Due to the exploratory and incremental nature of this research project, an exact comparison of these techniques under identical conditions could not be performed. Hence, we intend to test the computational tractability and accuracy of models that use different data discretization techniques under identical conditions.

We also intend to explore POMDPs and LIMIDs, and model decision support systems that can recommend decisions using utility and action nodes and reward functions. We believe that these capabilities will enhance the utility of probabilistic temporal reasoning techniques in the clinical domain. We will add new features to Projeny to support more learning and inference algorithms, and more modeling techniques such as POMDPs and LIMIDs. We hope to actively improve the tool with inputs from the larger open source community.

Finally, we intend to test the hyperglycemia and sepsis models with more rigor and with larger data sets, and validate them under stricter conditions. We intend to evaluate and refine these models for their appropriateness for use in a real clinical setting. We intend to apply these tools and techniques to other biomedical problems as well.

REFERENCES

[1] James G. An interview with Scott Adams; 2008. Cited on March 10, 2012. Available from: http://www.cbsnews.com/2102-505183_162-28540534.html.

[2] Shahar Y. Timing is everything: temporal reasoning and temporal data maintenance in medicine. In: Medicine, Lecture Notes In Artificial Intelligence 1620. Springer Verlag; 1999. p. 30–46.

[3] Beck JR, Pauker SG. The Markov process in medical prognosis. Med Decis Making. 1983;3(4):419–458.

[4] Christakis NA, Lamont EB. Extent and determinants of error in doctors' prognoses in terminally ill patients: prospective cohort study. BMJ. 2000 Feb;320(7233):469–472.

[5] Alexander M. Bias and asymmetric loss in expert forecasts: a study of physician prognostic behavior with respect to patient survival. J Health Econ. 2008 Jul;27(4):1095–1108.

[6] Casscells W, Schoenberger A, Graboys TB. Interpretation by physicians of clinical laboratory results. N Engl J Med. 1978 Nov;299(18):999–1001.

[7] Aronsky D, Fiszman M, Chapman WW, Haug PJ. Combining decision support methodologies to diagnose pneumonia. In: Proc AMIA Annu Symp; 2001. p. 12–16.

[8] Lagor C, Aronsky D, Fiszman M, Haug PJ. Automatic identification of patients eligible for a pneumonia guideline: comparing the diagnostic accuracy of two decision support models. Stud Health Technol Inform. 2001;84(Pt 1):493–497.

[9] Chapman WW, Fizman M, Chapman BE, Haug PJ. A comparison of classification algorithms to automatically identify chest X-ray reports that support pneumonia. J Biomed Inform. 2001 Feb;34(1):4–14.

[10] Pyrros A, Nikolaidis P, Yaghmai V, Zivin S, Tracy JI, Flanders A. A Bayesian approach for the categorization of radiology reports. Acad Radiol. 2007 Apr;14(4):426–430.

[11] Detrano R, Yiannikas J, Salcedo EE, Rincon G, Go RT, Williams G, et al. Bayesian probability analysis: a prospective demonstration of its clinical utility in diagnosing coronary disease. Circulation. 1984 Mar;69(3):541–547.

[12] Jovin GH, Vasilescu C, Popescu I, Cizma P. The causal diagnosis of obstructive jaundice. A Bayesian approach. Med Interne. 1988;26(3):213–220.

[13] Edwards FH, Schaefer PS, Cohen AJ, Bellamy RF, Thompson L, Graeber GM, et al. Use of artificial intelligence for the preoperative diagnosis of pulmonary lesions. Ann Thorac Surg. 1989 Oct;48(4):556–559.

[14] Gurney JW, Lyddon DM, McKay JA. Determining the likelihood of malignancy in solitary pulmonary nodules with Bayesian analysis. Part II. Application. Radiology. 1993 Feb;186(2):415–422.

[15] Andreassen S, Hovorka R, Benn J, Olesen KG, Carson ER. A model-based approach to insulin adjustment. In: Talmon MS, Hasman A, Fieschi M, J, editors. Third Conference on Artificial Intelligence in Medicine. Springer-Verlag; 1991. p. 239–248.

[16] Horvitz E, Seiver A, Service CC. Time-critical action: representations and application. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Providence, RI. Morgan Kaufmann; 1997. p. 250–257.

[17] Arroyo-Figueroa G, Sucar L. A temporal Bayesian network for diagnosis and prediction. In: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99). San Francisco, CA: Morgan Kaufmann; 1999. p. 13–20.

[18] Lucas PJF, van der Gaag LC, Abu-Hanna A. Bayesian networks in biomedicine and health-care. Artif Intell Med. 2004 Mar;30(3):201–214.

[19] Jackson CH, Sharples LD. Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients. Stat Med. 2002 Jan;21(1):113–128.

[20] Satten GA, Longini Jr IM. Markov chains with measurement error: estimating thetrue'course of a marker of the progression of human immunodeficiency virus disease. Applied Statistics. 1996;p. 275–309.

[21] Murphy KP. Dynamic bayesian networks: representation, inference and learning. PhD thesis. U.C.Berkeley; 2002.

[22] Yamanaka N, Okamoto E, Kuwata K, Tanaka N. A multiple regression equation for prediction of posthepatectomy liver failure. Ann Surg. 1984 Nov;200(5):658–663.

[23] Ohno-Machado L. Sequential use of neural networks for survival prediction in AIDS. Proc AMIA Annu Fall Symp. 1996;p. 170–174.

[24] Baghdadi G, Nasrabadi AM. Controlling blood glucose levels in diabetics by neural network predictor. Conf Proc IEEE Eng Med Biol Soc. 2007;2007:3216–3219.

[25] Harrell F Jr, Lee KL, Califf RM, Pryor DB, Rosati RA. Regression modelling strategies for improved prognostic prediction. Stat Med. 1984;3(2):143–152.

[26] Correa M, Bielza C, Pamies-Teixeira J. Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process. Expert Syst Appl. 2009 April;36:7270–7279.

[27] Lee SM, Abbott P, Johantgen M. Logistic regression and Bayesian networks to study outcomes using large data sets. Nursing Research. 2005;54(2):133.

[28] Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of Clinical Epidemiology. 1996;49(11):1225–1231.

[29] Uusitalo L. Advantages and challenges of Bayesian networks in environmental modelling. Ecological Modelling. 2007;203(3-4):312–318.

[30] Combi C, Shahar Y. Temporal reasoning and temporal data maintenance in medicine: issues and challenges. Comput Biol Med. 1997 Sep;27(5):353–368.

[31] Andreassen S, Jensen FV, Olesen KG. Medical expert systems based on causal probabilistic networks. Int J Biomed Comput. 1991;28(1-2):1–30.

[32] Van den Berghe G. Beyond diabetes: saving lives with insulin in the ICU. Int J Obes Relat Metab Disord. 2002;26 Suppl 3:S3–8.

[33] McCowen KC, Malhotra A, Bistrian BR. Stress-induced hyperglycemia. Crit Care Clin. 2001 Jan;17(1):107–124.

[34] Pittas AG, Siegel RD, Lau J. Insulin therapy and in-hospital mortality in critically ill patients: systematic review and meta-analysis of randomized controlled trials. JPEN J Parenter Enteral Nutr. 2006;30(2):164–172.

[35] Griesdale DEG, de Souza RJ, van Dam RM, Heyland DK, Cook DJ, Malhotra A, et al. Intensive insulin therapy and mortality among critically ill patients: a meta-analysis including NICE-SUGAR study data. CMAJ. 2009 Apr;180(8):821–827.

[36] Treggiari MM, Karir V, Yanez ND, Weiss NS, Daniel S, Deem SA. Intensive insulin therapy and mortality in critically ill patients. Crit Care. 2008;12(1):R29.

[37] Morris AH, Orme Jr J, Truwit JD, Steingrub J, Grissom C, Lee KH, et al. A replicable method for blood glucose control in critically ill patients. Critical Care Medicine. 2008;36(6):1787.

[38] de Oliveira CF. Early goal-directed therapy in treatment of pediatric septic shock. Shock. 2010 Sep;34 Suppl 1:44–47.

[39] Rollins G. First-ever sepsis guidelines call for aggressive recognition and early goal-directed therapy. Rep Med Guidel Outcomes Res. 2004 Apr;15(7):1–2, 5–6.

[40] Rivers E, Nguyen B, Havstad S, Ressler J, Muzzin A, Knoblich B, et al. Early goal-directed therapy in the treatment of severe sepsis and septic shock. N Engl J Med. 2001 Nov;345(19):1368–1377.

[41] Rivers EP, McIntyre L, Morro DC, Rivers KK. Early and innovative interventions for severe sepsis and septic shock: taking advantage of a window of opportunity. CMAJ. 2005 Oct;173(9):1054–1065.

[42] Murphy KP. BNT - Bayes Net Toolbox;. Cited on March 13, 2009. Available from: http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html.

[43] Müller S. JMatLink;. Cited on March 13, 2009. Available from: http://jmatlink.sourceforge.net.

[44] Hsu, et al.. BNJ - Bayesian Network Tools in Java;. Cited on March 13, 2009. Available from: http://bnj.sourceforge.net/.

[45] O'Connor MJ, Tu SW, Musen MA. The Chronus II temporal database mediator. In: Proc AMIA Annu Symp; 2002. p. 567–571.

[46] Lin Y. Two theories of time. Journal of Applied Non-Classical Logics. 1999;1(1).

[47] Keravnou ET. Temporal reasoning in medicine. Artificial Intelligence in Medicine. 1996;8(3):187–191.

[48] Allen JF. Maintaining knowledge about temporal intervals. Commun ACM. 1983 nov;26(11):832–843.

[49] Halpern JY, Shoham Y. A propositional modal logic of time intervals. Journal of the ACM (JACM). 1991;38(4):935–962.

[50] Reiter R. Natural actions, concurrency and continuous time in the situation calculus. do (c; s);1:2.

[51] Shanahan M. Representing continuous change in the event calculus. In: Proceedings ECAI. vol. 90; 1990. p. 598–603.

[52] Sandewall E. Combining logic and differential equations for describing real-world systems. In: Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann Pub; 1989. p. 412–420.

[53] Nodelman U, Shelton CR, Koller D. Continuous time Bayesian networks. In: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence; 2002. p. 378–387.

[54] Nodelman U, Shelton CR, Koller D. Learning continuous time Bayesian networks. In: Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03); 2003. p. 451–8.

[55] Andersson S, Madigan D, Perlman MD. A characterization of Markov equivalence classes for acyclic digraphs. Annals of Statistics. 1995;25:505–541.

[56] Mccarthy J, Hayes PJ. Some philosophical problems from the standpoint of artificial intelligence. In: Machine Intelligence. Edinburgh University Press; 1969. p. 463–502.

[57] McCarthy J. Situations, actions, and causal laws. Stanford Univ Calif Dept of Computer Science; 1963.

[58] Reiter R. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy. 1991;p. 359–380.

[59] Kowalski R, Sergot M. A logic-based calculus of events. New Generation Computing. 1986;4(1):67–95.

[60] Sandewall E. Features and Fluents: The representation of knowledge about dynamical systems. Oxford University Press, USA; 1994.

[61] Hölldobler S, Thielscher M. Computing change and specificity with equational logic programs. Annals of Mathematics and Artificial Intelligence. 1995;14(1):99–133.

[62] Thielscher M. From situation calculus to fluent calculus: state update axioms as a solution to the inferential frame problem. Artificial Intelligence. 1999;111(1):277–299.

[63] Lamport L. The temporal logic of actions. ACM Transactions on Programming Languages and Systems (TOPLAS). 1994;16(3):872–923.

[64] Gelfond M, Lifschitz V. Representing action and change by logic programs. Journal of Logic Programming. 1993;17(2):301–321.

[65] Vilain M, Kautz H. Constraint propagation algorithms for temporal reasoning. In: Proceedings of the Fifth National Conference on Artificial Intelligence; 1986. p. 377–382.

[66] Nebel B, Bürckert HJ. Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra. Journal of the ACM (JACM). 1995;42(1):43–66.

[67] Mackworth AK. Consistency in networks of relations. Artificial Intelligence. 1977;8(1):99–118.

[68] Montanari U. Networks of constraints: Fundamental properties and applications to picture processing. Information Sciences. 1974;7:95132.

[69] Galton A. Reified temporal theories and how to unreify them. In: Proceedings of IJCAI 91. Citeseer; 1991. p. 1177–1182.

[70] Kumar V. Algorithms for constraint-satisfaction problems: A survey. AI Magazine. 1992;13(1):32.

[71] Dechter R, Meiri I, Pearl J. Temporal constraint networks. Artificial Intelligence. 1991;49:61–95.

[72] Navarrete I, Marin R. Qualitative temporal reasoning with points and durations. In: International Joint Conference on Artificial Intelligence. vol. 15; 1997. p. 1454–1459.

[73] Kautz HA, Ladkin PB. Integrating metric and qualitative temporal reasoning. In: Proceedings of AAAI-91. Citeseer; 1991. p. 241–246.

[74] Meiri I. Combining qualitative and quantitative constraints in temporal reasoning. Artificial Intelligence. 1996;87(1-2):343–385.

[75] Wetprasit R, Sattar A. Temporal reasoning with qualitative and quantitative information about points and durations. In: Proceedings of the National Conference on Artificial Intelligence; 1998. p. 656–664.

[76] Chittaro L, Montanari A. Temporal representation and reasoning in artificial intelligence: issues and approaches. Annals of Mathematics and Artificial Intelligence. 2000;28(1):47–106.

[77] Wiederhold G, Fries JF, Weyl S. Structured organization of clinical databases. In: Proceedings of the National Computer Conference. vol. 44; 1975. p. 479–485.

[78] Blum R. Discovery and representation of causal relationships from a large time-oriented clinical database: the RX project. Dissertation Abstracts International Part B: Science and Engineering[DISS ABST INT PT B- SCI & ENG],. 1982;42(11):1982.

[79] Kahn M, Fagan L, Tu S. Extensions to the time-oriented database model to support temporal reasoning in medical expert systems. Methods of Information in Medicine. 1991;30(1):4.

[80] Pinciroli F, Combi C, Pozzi G. Object-orientated DBMS techniques for time-orientated medical record. Informatics for Health and Social Care. 1992;17(4):231–241.

[81] Snodgrass RT. Developing time-oriented database applications in SQL. Morgan Kaufmann; 2000.

[82] Das A, Musen M. A temporal query system for protocol-directed decision support. Methods of Information in Medicine. 1994;33(4):358.

[83] Das A, Tu S, Purcell G, Musen M. An extended SQL for temporal data management in clinical decision-support systems. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. American Medical Informatics Association; 1992. p. 128.

[84] Das AK, Shahar Y, Tu SW, Musen MA. A temporal-abstraction mediator for protocol-based decision-support systems. Proc Annu Symp Comput Appl Med Care. 1994;p. 320–324.

[85] Combi C, Pinciroli F, Cavallaro M, Cucchi G. Querying temporal clinical databases with different time granularities: the GCH-OSQL language. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. American Medical Informatics Association; 1995. p. 326.

[86] Combi C, Missora L, Pinciroli F. Supporting temporal queries on clinical relational databases: the S-WATCH-QL language. In: Proceedings of the AMIA Annual Fall Symposium. American Medical Informatics Association; 1996. p. 527.

[87] Snodgrass RT. The TSQL2 temporal query language. Kluwer Academic Pub; 1995.

[88] Date C, Darwen H. Temporal Data and the Relational Model. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2002.

[89] McBrien P, Owens R, Gabbay D, Niezette M, Wolper P. TEMPORA: a temporal database transaction system. In: IEE Colloquium on Temporal Reasoning; 1990. p. 4.

[90] Tu SW, Kahn MG, Musen MA, Fagan LM, Ferguson JC. Episodic skeletal-plan refinement based on temporal data. Commun ACM. 1989 Dec;32(12):1439–1455.

[91] Box GEP, Jenkins GM, Reinsel GC. Time series analysis. Holden-day San Francisco; 1970.

[92] Brockwell PJ, Davis RA. Time series: theory and methods. Springer; 1991.

[93] Hamilton JD. Time series analysis. Princeton Univ Pr; 1994.

[94] Kedem B, Fokianos K. Regression models for time series analysis. John Wiley and Sons; 2002.

[95] Mukherjee S, Osuna E, Girosi F. Nonlinear prediction of chaotic time series using support vector machines. In: Proc. of IEEE 1997 NNSP Workshop; 1997. p. 511 – 520.

[96] Ruping S. SVM kernels for time series analysis. In: LLWA. Citeseer; 2001. p. 43–50.

[97] Cao L, Tay FEH. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks. 2003;14(6):1506–1518.

[98] Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural computation. 1989;1(2):270–280.

[99] Dehaene S, Changeux J, Nadal J. Neural networks that learn temporal sequences by selection. Proceedings of the National Academy of Sciences. 1987;84(9):2727.

[100] Jang JSR. Self-learning fuzzy controllers based on temporal backpropagation. IEEE Transactions on Neural Networks. 1992;3(5):714–723.

[101] Dayhoff JE, DeLeo JM. Artificial neural networks. CA A Cancer Journal for Clinicians;91(S8):1615–1635.

[102] Hart A, Wyatt J. Evaluating black-boxes as medical decision aids: issues arising from a study of neural networks. Med Inform (Lond). 1990;15(3):229–36. 0307-7640 (Print)Journal Article.

[103] Lauritzen S, Spiegelhalter D. Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society Series B (Methodological). 1988;50(2):157–224.

[104] Berkson J. Limitations of the application of fourfold table analysis to hospital data. Biometrics Bulletin. 1946;2(3):47–53.

[105] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Readings in Speech Recognition. 1990;53(3):267–296.

[106] Roweis S, Ghahramani Z. A unifying review of linear Gaussian models. Neural Computation. 1999;11(2):305–345.

[107] Minka TP. From hidden Markov models to linear dynamical systems. Department of Electrical Engineering and Computer Science, MIT, Cambridge, Mass, USA; 1999.

[108] Monahan GE. A survey of partially observable Markov decision processes: theory, models, and algorithms. Management Science. 1982;28(1):1–16.

[109] Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. Artificial Intelligence. 1998;101(1-2):99–134.

[110] Cassandra AR. Exact and approximate algorithms for partially observable Markov decision processes. Brown University; 1998.

[111] Nilsson D, Lauritzen SL. Evaluating influence diagrams using LIMIDs. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence. Citeseer; 2000. p. 436–445.

[112] Lauritzen SL, Nilsson D. Representing and solving decision problems with limited information. Management Science. 2001;47(9):1235–1251.

[113] van Gerven MAJ, Díez FJ, Taal BG, Lucas PJF. Selecting treatment strategies with dynamic limited-memory influence diagrams. Artificial Intelligence in Medicine. 2007;40(3):171–186.

[114] Murphy K. A brief introduction to graphical models and Bayesian networks; 1998. Cited on November 30, 2009. Available from: http://people.cs.ubc.ca/~murphyk/Bayes/bnintro.html#infer.

[115] Zweig G. A forward-backward algorithm for inference in Bayesian networks and an empirical comparison with HMMs [Master's thesis]. U.C. Berkeley; 1996.

[116] Zhang NL, Poole D. Exploiting causal independence in Bayesian network inference. Arxiv preprint cs/9612101. 1996;.

[117] Boyen X, Koller D. Tractable inference for complex stochastic processes. In: Proc. UAI. vol. 98; 1998. p. 33–42.

[118] Murphy K, Weiss Y. The factored frontier algorithm for approximate inference in DBNs. In: Proc. UAI; 2001. p. 378–385.

[119] Pearl J, Shafer G. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann San Mateo, CA; 1988.

[120] Bar-Shalom Y, Li XR. Estimation and tracking - principles, techniques, and software. Norwood, MA: Artech House, Inc, 1993.; 1993.

[121] Minka TP. A family of algorithms for approximate Bayesian inference. Massachusetts Institute of Technology; 2001.

[122] Casella G, Robert CP. Rao-Blackwellisation of sampling schemes. Biometrika. 1996;83(1):81.

[123] Lauritzen SL. The EM algorithm for graphical association models with missing data. Computational Statistics & Data Analysis. 1995;19(2):191–201.

[124] Friedman N. The Bayesian structural EM algorithm. In: Proc. UAI. vol. 98; 1998. p. 129–138.

[125] Friedman N. Learning belief networks in the presence of missing values and hidden variables. In: Proceedings of the Fourteenth International Conference on Machine Learning; 1997. p. 125–133.

[126] Friedman N, Murphy K, Russell S. Learning the structure of dynamic probabilistic networks. In: Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI98). Citeseer; 1998. p. 139–147.

[127] Dagum P, Galper A. Forecasting sleep apnea with dynamic network models. In: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence; 1993. p. 64–71.

[128] Hernando ME, Gmez EJ, del Pozo F, Corcoy R. DIABNET: a qualitative model-based advisory system for therapy planning in gestational diabetes. Med Inform (Lond). 1996;21(4):359–374.

[129] Hernando ME, Gmez EJ, Corcoy R, del Pozo F. Evaluation of DIABNET, a decision support system for therapy planning in gestational diabetes. Comput Methods Programs Biomed. 2000 Jul;62(3):235–248.

[130] Leong TY. Multiple perspective dynamic decision making. Artificial Intelligence. 1998;105(1-2):209–261.

[131] Provan GM, Clarke JR. Dynamic network construction and updating techniques for the diagnosis of acute abdominal pain. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993;p. 299–307.

[132] Galn SF, Aguado F, Dez FJ, Mira J. NasoNet, modeling the spread of nasopharyngeal cancer with networks of probabilistic events in discrete time. Artif Intell Med. 2002 Jul;25(3):247–264.

[133] Sebastiani P, Mandl KD, Szolovits P, Kohane IS, Ramoni MF. A Bayesian dynamic model for influenza surveillance. Stat Med. 2006 Jun;25(11):1803–16; discussion 1817–25.

[134] Xiang Z, Minter RM, Bi X, Woolf PJ, He Y. miniTUBA: medical inference by network integration of temporal data using Bayesian analysis. Bioinformatics. 2007;23(18):2423.

[135] van Gerven MAJ, Taal BG, Lucas PJF. Dynamic Bayesian networks as prognostic models for clinical patient management. Journal of Biomedical Informatics. 2008;41(4):515–529.

[136] Charitos T, van der Gaag LC, Visscher S, Schurink KAM, Lucas PJF. A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients. Expert Systems with Applications. 2009;36(2):1249–1258.

[137] Langmead CJ. Generalized queries and Bayesian statistical model checking in dynamic Bayesian networks: Application to personalized medicine. In: Proc. of the 8th International Conference on Computational Systems Bioinformatics (CSB). Citeseer; 2009. p. 201–212.

[138] Peelen L, de Keizer NF, Jonge Ed, Bosman RJ, Abu-Hanna A, Peek N. Using hierarchical dynamic Bayesian networks to investigate dynamics of organ failure in patients in the Intensive Care Unit. J Biomed Inform. 2010 Apr;43(2):273–286.

[139] Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP system. J Med Syst. 1983 Apr;7(2):87–102.

[140] Clayton PD, Narus SP, Huff SM, Pryor TA, Haug PJ, Larkin T, et al. Building a comprehensive clinical information system from components. The approach at Intermountain Health Care. Methods Inf Med. 2003;42(1):1–7.

[141] Little RJA, Rubin DB. Statistical analysis with missing data. 1st ed. Wiley; 1987.

[142] Lin JH. Data preparation for clinical data mining in developing a problem list proposing system. The University of Utah; 2008.

[143] Lin JH, Haug PJ. Exploiting missing clinical data in Bayesian network modeling for predicting medical problems. Journal of Biomedical Informatics. 2008;41(1):1–14.

[144] Mitsa T. Temporal Data Mining. Chapman & Hall/CRC data mining and knowledge discovery series. Chapman & Hall/CRC; 2009.

[145] Warren Liao T. Clustering of time series data–a survey. Pattern Recognition. 2005;38(11):1857–1874.

[146] Bar-Joseph Z. Analyzing time series gene expression data. Bioinformatics. 2004;20(16):2493.

[147] Liu H, Hussain F, Tan CLIM, Dash M. Discretization: an enabling technique. Data Mining and Knowledge Discovery. 2002;6:393–423.

[148] Kotsiantis S, Kanellopoulos D. Discretization techniques: a recent survey. GESTS International Transactions on Computer Science and Engineering. 2006;32(1):47–58.

[149] Hulst J. Modeling physiological processes with dynamic Bayesian networks. Delft University of Technology; 2006.

[150] Lloyd SP. Least squares quantization in PCM. IEEE Transactions on Information Theory. 1982;28(2):129–137.

[151] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter. 2009;11(1):10–18.

[152] Fayyad UM, Irani KB. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. Morgan-Kaufmann; 1993. p. 1022–1027.

[153] Murphy K, et al. The bayes net toolbox for Matlab. Computing Science and Statistics. 2001;33(2):1024–1034.

[154] Nachimuthu SK. Projeny - Probabilistic Networks Generator in Java; 2010. Cited on March 13, 2009. Available from: http://projeny.sourceforge.net.

[155] Matlab; 2009. Cited on March 13, 2009. Available from: http://www.mathworks.com.

[156] Witten IH, Frank E. Data mining: practical machine learning tools and techniques. Morgan Kaufmann; 2005.

[157] Cho IS, Haug PJ. The contribution of nursing data to the development of a predictive model for the detection of acute pancreatitis. Stud Health Technol Inform. 2006;122:139–142.

[158] Van den Berghe G, Schetz M, Vlasselaers D, Hermans G, Wilmer A, Bouillon R, et al. Clinical review: intensive insulin therapy in critically ill patients: NICE-SUGAR or Leuven blood glucose target? J Clin Endocrinol Metab. 2009 Sep;94(9):3163–3170.

[159] Van den Berghe G. Insulin therapy for the critically ill patient. Clin Cornerstone. 2003;5(2):56–63.

[160] Griesdale DEG, de Souza RJ, van Dam RM, Heyland DK, Cook DJ, Malhotra A, et al. Intensive insulin therapy and mortality among critically ill patients: a meta-analysis including NICE-SUGAR study data. CMAJ. 2009 Apr;180(8):821–827.

[161] Morris AH. Treatment algorithms and protocolized care. Curr Opin Crit Care. 2003 Jun;9(3):236–240.

[162] Morris AH. Developing and implementing computerized protocols for standardization of clinical decisions. Ann Intern Med. 2000 Mar;132(5):373–383.

[163] McDonald CJ, Overhage JM, Tierney WM, Abernathy GR, Dexter PR. The promise of computerized feedback systems for diabetes care. Ann Intern Med. 1996 Jan;124(1 Pt 2):170–174.

[164] Goldberg PA, Siegel MD, Sherwin RS, Halickman JI, Lee M, Bailey VA, et al. Implementation of a safe and effective insulin infusion protocol in a medical intensive care unit. Diabetes Care. 2004 Feb;27(2):461–467.

[165] Wong A, Al-Sheikh Y, Warner H, Haug P, Sward K, Morris A. Evaluation and comparison of IV insulin-treatment protocols using data from critically ill patients in the ICU. In: AMIA... Annual Symposium proceedings/AMIA Symposium. AMIA Symposium; 2008. p. 1178.

[166] Sward K, Orme J Jr, Sorenson D, Baumann L, Morris AH, RCCCRI. Reasons for declining computerized insulin protocol recommendations: application of a framework. J Biomed Inform. 2008 Jun;41(3):488–497.

[167] Youden W. Index for rating diagnostic tests. Cancer. 1950;3(1):32–35.

[168] Metz CE, Herman BA, Roe CA. Statistical comparison of two ROC-curve estimates obtained from partially-paired datasets. Medical Decision Making. 1998;18(1):110–121.

[169] Dremsizov TT, Kellum JA, Angus DC. Incidence and definition of sepsis and associated organ dysfunction. Int J Artif Organs. 2004 May;27(5):352–359.

[170] Moerer O, Quintel M. Sepsis in adult patients - definitions, epidemiology and economic aspects. Internist (Berl). 2009 Jul;50(7):788, 790–4, 796–8.

[171] Bone RC, Balk RA, Cerra FB, Dellinger RP, Fein AM, Knaus WA, et al. Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. The ACCP/SCCM Consensus Conference Committee. American College of Chest Physicians/Society of Critical Care Medicine. Chest. 1992 Jun;101(6):1644–1655.

[172] Morrison AM. Receiver Operating Characteristic (ROC) Curve PreparationA Tutorial. Boston: Massachusetts Water Resources Authority Report ENQUAD. 2005;20(5).