Statistical Timeline Analysis for Electronic Health Records

By

Jeremy C. Weiss

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Computer Science)

at the

UNIVERSITY OF WISCONSIN–MADISON

2014

Date of final oral examination: 04/14/2014

The dissertation is approved by the following members of the Final Oral Committee:
C. David Page, Professor, Computer Science
Mark Craven, Professor, Computer Science
Sriraam Natarajan, Assistant Professor, Informatics and Computing
Xiaojin Zhu, Associate Professor, Computer Science
Deane Mosher, Professor, Biomolecular Chemistry

To my family and the future, the anchors of my present and leaders of my past.

Acknowledgments

First and foremost, I want to thank my advisor David Page, who has mentored me tirelessly. David incited in me a dedication to the science, guiding me through difficult problems with emphasis on the multi-armed approach to modeling and interpretation and with reinforcement of the pillars of applied machine learning. His research platform provided a strong foundation: from support of focused research time to freedom in methodology for the medical machine learning tasks. In terms of problem solving, he liked to assert that the simplest approach works, leaving me to pursue its success conditions and to investigate alternative solutions upon failure. Investigation of the alternative solution led me to the continuous-time framework I adopted in this thesis. With his guidance drawing from his depth of knowledge and his ability to frame outstanding problems, I have a greater appreciation for computer science as a whole and a better perspective on my future and our field's.

I want to thank my committee members for their support, guidance, and for collectively examining my project from their broad ranges of expertise to make it stronger. Professor Mark Craven introduced the continuous-time modeling approach to me, which grew into a key theme of my work. His ability to pose questions that require you to explain the heart of the matter is worth emulating. Professor Sriraam Natarajan has consistently provided the spark to dive into new problems, critically analyzing them while drawing from strengths of existing yet diverse approaches. His excitement and love for machine learning is infectious and a great asset. Professor Deane Mosher has provided unparalleled support individually and as Director of the MD-PhD program and was a key factor in my decision to come to Madison and in my success here. Professor Xiaojin Zhu was the first to help me tie together the mathematical breadth of machine learning with the notion of machine learning as a computational tool for artificial intelligence.

I want to thank the University of Wisconsin-Madison, the University of Pennsylvania, and the Lakeside School for situating me in environments conducive to learning and the academic freedom to explore. The Medical Scientist Training Program and Computer Sciences Department underlie my successes and I could not be at this point without them. My funding sources entrusted me with time; time to get lost in the abyss of technological knowledge so that I could return with more tools at my belt. These include the Institute for Clinical and Translational Research and Computation and Informatics in Biology and Medicine Training Programs, as well as the Machine Learning for Identifying Adverse Drug Event R01GM097618.

I want to thank my collaborators and co-authors, specifically Peggy Peissig, Catherine McCarty, Michael Caldwell, Bess Berg, Jie Liu, Kendrick Boyd, and Finn Kuusisto, without whom this work would not have been possible. My thanks also go to the many more who have helped shape my research ideas.

I want to thank my friends for great times and meaningful conversations, broadening my knowledge of the shape of the world, computer wherewithal, and cultural endemism.

I especially want to thank my family–Noel, Chu, and Jessica–for shaping the world around me and providing for me throughout the process. To those in North Seattle and Camarillo, you are on my mind and your spirits live on in me.

DISCARD THIS PAGE

TABLE OF CONTENTS

DISCARD THIS PAGE

LIST OF TABLES

Table Page

DISCARD THIS PAGE

LIST OF FIGURES

Abstract

Electronic Health Records (EHRs) now hold over 50 years of recorded patient information and, with increased adoption and high levels of population coverage, are becoming foci of public health analyses. The structure of EHR patient data limits existing clinical study paradigms, which fail to effectively capture the relational, temporal, and intermittent data characteristics. This dissertation develops statistical timeline analysis (STA), a set of algorithms that extend existing modeling approaches to address EHR data challenges.

Statistical timeline analysis models EHR data as patient-specific, relational timelines, where measurements and events occur in continuous-time instead of at fixed intervals. First, we adopt a relational forest algorithm and show improved performance at heart attack prediction compared to analogous non-relational algorithms. Then we turn to richer timeline models: continuous-time Bayesian networks (CTBNs), which model dependencies in rate among discrete variables over continuous time. We introduce partition-based CTBNs, a generalization that alleviates the exponential space constraints of CTBNs yet maintains the ability to model complex dependencies. We then develop a multiplicative forest learning algorithm with space linear in the number of forest splits that efficiently maximizes the partition-based CTBN likelihood.

To address CTBN inference challenges, we identify a general method for the improvement of sequential importance samples. Our method reduces sample weight variance by an order of magnitude, yielding a better approximation of the posterior distribution.

We also study point processes, which avoid CTBN inference challenges altogether. We show that the multiplicative forest learning algorithm applies and improves upon existing learning algorithms both in modeling dependences and as extracted features for forecasting heart attacks.

Finally we turn to attributable risk. The clinical study paradigm focuses on population-average changes in risk. However, the average outcomes of such studies are then applied to individuals when the application of the individual outcome is more appropriate. We show that individualized-risk modeling improves average individual outcomes and provides evidence of the EHR as an effective source for modeling individualized attributable risks.

Our contributions to statistical timeline analysis show algorithmic and performance improvements that address EHR data challenges. We expect further research combining these ideas to improve clinical understanding and patient care.

1 Introduction

Alongside initiatives to increase the availability of Electronic Health Record (EHR) information through the Affordable Care Act and initiatives to deploy increasing numbers of mobile health (mHealth) studies, analysts of clinical data are discovering limitations of existing methodologies and are developing machine learning methods to meet their data challenges (Jha, 2010; Kay et al., 2011). These trends signal a shift in the collection and recording of patient health data, with the EHR becoming a primary data source for clinical analysis.

As the adoption of the EHR as a data source increases, novel methods in machine learning and biostatistics for analyzing EHR data are needed to derive utility in the form of clinical findings. This thesis seeks to describe the challenges of using EHR data to produce clinical findings and develop a set of algorithms to answer clinically important queries from such data. A few challenges in analyzing EHR data, which we describe in detail in Section 2.1, include the effective and efficient use of large-width tables, the ability to capture temporal effects, the use of heterogeneous data sources, and the ability to leverage relational database structure.

We should not be deterred by these challenges because the continued use specifically of EHR data is certain to drive new clinical findings. For one, other existing methods driving clinical findings–clinical studies and in particular randomized trials–cannot scale to address each exposure-outcome pair of interest. With thousands of diagnoses, e.g., ICD-10 codes, and orders of magnitude more measurements of potential risk factors, clinical studies must limit their focus to common outcomes or treatments with large benefit. Analyses from EHR data do not have such limitations. Furthermore, EHR data may provide a richer patient profile than clinical study data where potential risks measured are pre-specified in the study protocol. Thus, the use of EHR data can lead to improved predictions and better disease characterization per dollar spent. Clinical trial findings can also become "stale", *i.e.*, the results may not apply well to patients in the future because medical care protocol has changed. A re-analysis from EHR data can update the clinical recommendations without additional intervention. Finally, EHR data hold records on large and diverse sets of people, increasing statistical power to detect clinical findings and to characterize heterogeneity among subpopulations.

This thesis develops an analysis of EHR data to answer two types of tasks: first, the prediction or forecasting of a patient outcome, and second, the estimation of the risk of an outcome attributable to an exposure or treatment. To do so, we propose the framework of statistical timeline analysis (STA), which places emphasis on the temporal nature of clinical events, on EHR data associated specifically with patient identifiers, and on a probabilistic regime for answering queries.

1.1 Clinical Motivation

To motivate the usefulness of clinical outcome prediction and risk attribution from EHR data, let us consider an example of a physician note.

> A 63 year old white male comes to clinic complaining of one month's duration of chest pain after non-strenuous exercise such as climbing 2 flights of stairs to his apartment. The pain is diffuse in the left front of the chest, rated an 8/10, and is relieved by rest and sitting or lying down. Past medical history is significant for high blood pressure and high cholesterol. The patient is a current smoker with a history of 20 pack-years.

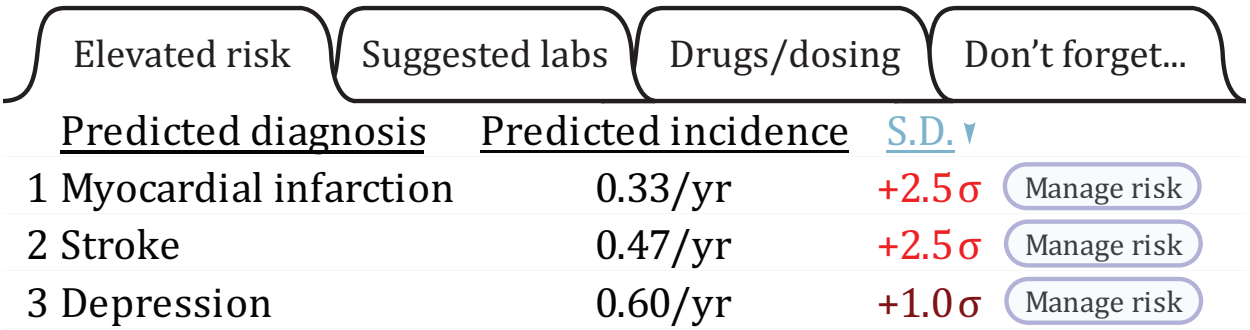

Figure 1.1: This figure shows a possible future EHR interface for the physician that includes machine learning predictions for the current patient. The diagram shows model results suggesting that the patient is at elevated risk for specific diagnoses. It depicts a tabbed environment, where the machine learning system also provides optimal drug regimens, recommends the collection of additional health information such as laboratory assays, and reminds physicians of steps involved in providing continuing care.

This clinic note briefly describes the chief complaint (CC), history of present illness (HPI), and past medical history (PMH). In just a short note, the pertinent features for establishing the abnormal condition let the physician construct a differential diagnosis. Following the history (CC, HPI, and PMH), a physical exam is performed. In broad terms, the physician initially follows a data acquisition phase, followed by an exploratory or confirmatory phase, finally leading to a diagnosis, prognosis and treatment.

At each phase of this process, clinical findings guide the physician. Classifiers can help the physician weigh likely diagnoses and models can help characterize likely underlying disease processes. Risk scores can help the physician understand the expected benefit of treatment choices and the range of likely outcomes. Computerized alerts can remind the physician to renew or terminate drug prescriptions, as illustrated in Figure 1.1. At each decision point, a physician armed with such information can make more informed decisions that may improve the average outcome of patients.

Here is another use case. As medical students learn to hone their clinical acumen, two of the most common questions posed to them are: (1) what is the most likely diagnosis, and (2) what is the next step? Again, the answers come back to the use of clinical findings to guide medical decision-making.

In terms of applied machine learning, these two questions translate to (1) prediction and (2) risk attribution. The "most likely diagnosis" question asks for a classification and is typically followed by a question to list alternative but less probable diagnoses. The "next step" question typically requests the student to identify missing features that would help improve the certainty about the prediction of one or a few diseases. Our goal is to show that machine learning can provide patient-specific answers to these questions directly from EHR data. In the next section we describe methodologies to provide answers to these two questions in more detail.

1.2 Clinically-Applied Machine Learning

Both the "prediction" and the "risk attribution" tasks have devoted fields of study. We introduce these fields, discuss their limitations in addressing EHR data challenges, and describe how statistical timeline analysis

improves upon them.

Clinical forecasting

For prediction, machine learning and statistics have methods for classification and regression. However, existing methods do not meet the challenges existing in EHR data, such as its temporal, relational, and intermittent-arrival characteristics. Chapter 2 discusses these challenges at length, as they guide our choice of algorithm.

Specifically, work in Chapter 3 addresses the relational representation of EHR data. For prediction of primary myocardial infarctions (MIs), the work shows that a state-of-the-art relational forest learning method outperforms analogous algorithms that do not leverage the relational representation.

However, the relational forest algorithm does not model the temporal relationships beyond logical event time comparisons. Chapters 4, 5, and 6 discuss continuous-time Bayesian networks (CTBNs) and point processes, both of which model the rates of events over continuous time. As described in Section 2.1, continuous-time modeling is important for patient timelines, because observations of medical encounters do not arrive at regular intervals, an assumption that pervades existing clinical analyses.

In particular, CTBNs model events for every time t, but because EHR data describe events at time points, inferences about the events between time points becomes necessary. An improvement to existing CTBN inference methods based on the incorporation of a rejection sampling step in sequential importance sampling is the subject of Chapter 5.

Point processes, on the other hand, do not require interpolative inference and thus can scale to problems involving many event types. This comes at the cost of making a closed-world assumption, namely, that the observation of events define the occurrence of the event and the absence of observation means the event does not occur.

These three chapters lay the foundation for statistical timeline analysis: they show that, by representing EHR data as timelines, we can learn models that effectively describe medical event dependencies, which can then be used to improve forecasts about patient outcomes.

Risk attribution

For risk attribution, clinical studies from biostatistics and epidemiology are well-suited to answer such questions from data that are produced according to specific study designs. Randomized controlled trials (RCTs), cohort studies, and case-control studies all seek to approximate the risk attribution of an exposure for a disease of interest. Their main outcome, the average treatment effect (ATE), or its fractional relative, the relative risk (RR), describes the expected reduction in risk in the studied population given exposure.

RCTs have the important characteristic that confounding effects are mitigated by the randomization procedure, so an unbiased estimate of the ATE can be computed. They are, however, often impractical, infeasible, or unethical, so cohort and case-control studies are used in an attempt to mimic their outcomes. A variety of approaches are used, including controlling for confounders, propensity scoring, or inverse-probability-of-treatment weighting (Prentice, 1976; Austin, 2011; Rosenbaum & Rubin, 1983; Robins et al., 2000).

One critical drawback of all these methods is that they seek to calculate the average treatment effect, when most applications of risk attribution really desire the individualized treatment effect (ITE). The ITE provides the effect per individual instead of a population-level effect, and information about future individuals can be leveraged in determining optimal treatment choices. The predominant method for providing individualized

treatment effects from RCT-style analyses is through subgroup analyses, called heterogeneity of treatment effect (HTE) analysis. Our work in Chapter 7 suggests that, given clinical interest in the ITE, the procedure of finding the ATE and performing a secondary HTE analysis is indirect and possesses generalizability limitations that are not applicable to our method of ITE estimation. Furthermore, the use of EHR data introduces challenges for clinical study analyses; machine learning methods may be better suited to such data.

The field of machine learning focuses less on risk attribution estimation directly, but the closely related analysis of model interpretability is emphasized. Sometimes, interpretations of models directly answer the outcome of interest, *e.g.* the exposure coefficient of the logistic regression as the log odds ratio. The predominant model used in our work is the multiplicative forest, which is more challenging to interpret. We describe the forest models in detail in Chapters 4 and 6; here, we say that the forests determine if there is a dependency between two events, and the magnitude of the dependency can be calculated given the state of other pertinent events (*i.e.*, effect modifiers). We explore boosted forests in Chapter 7 for ITE estimation as an alternative to model inspection for interpretable results.

The use of EHR data and the development of algorithms that address the prediction and risk attribution questions bring us to the thesis statement.

1.3 Thesis Statement

In this thesis, we develop statistical timeline analysis, a set of algorithms that extend existing modeling approaches, to learn from Electronic Health Records data. We demonstrate that *statistical timeline analysis has utility in capturing the temporal and relational characteristics of population data and can be used to discover patient-specific clinical findings*.

2 Background

This chapter provides descriptions and definitions of foundational fields and tools for ensuing chapters. We motivate the use of Electronic Health Record data, introduce Statistical Relational Learning, describe two timeline models–the continuous-time Bayesian network and the point process–and finish with a brief review of clinical study methodology.

2.1 The Electronic Health Record as a Data Source

Electronic Health Records (EHRs) are an emerging data source of great potential use in disease prevention, diagnosis and treatment. An EHR tracks health trajectories of its patients through time for cohorts with stable populations (Figure 2.1). As of yet they have been used primarily as a data warehouse for patient health queries, rather than as a source for population-level risk assessment and prevention. This trend is changing, however, as exemplified by the Heritage Health Prize contest, which uses medical claims data to predict future hospitalization Heritage Provider Network (2011). In our work we will suggest that the emergence of the EHR as the new data source for population health analyses may allow us answer individualized clinical questions, as shown in Figure 2.2.

Findings discovered from EHR data can improve patient care, for example, by providing prompts to clinicians such as, "your patient is at high risk for an MI and is not currently on an aspirin regimen." Second, models build from EHRs can be inspected in order to identify surprising connections, such as a correlation between the outcome and the use of certain drugs, which might in turn provide important clinical insights. Third, findings derived from EHRs can be used in research to identify potential subjects for research studies. For example, if we want to test a new therapy for its ability to prevent an event such as MI, it would be most instructive to test it in a population of high-risk subjects.

EHR data present significant challenges to current machine learning methodology. If we hope to augment traditional clinical study analyses, we must be able to effectively address these challenges. A few of them are listed below.

- **Incomplete data.** Data typically consist of a variety of incomplete information: patient medical history, procedures history, family history, demographic information, self-reported questionnaire answers, lab tests, and genetic information. There is also provider information: location of services, pharmacy records, and insurance records. Integrating the variety of information available in an EHR is challenging, doubly so given that the extraction of insightful results comes from incomplete records.
- **EHR size.** EHRs include patients (thousands), providers (thousands), diagnoses and drugs (thousands), and in the near future genetic biomarkers (millions) and sequence data. Identifying complex relationships between these entities in a computationally efficient manner can be problematic.
- **Timestamps.** The trajectories of medical events are highly non-uniform; most medical encounters occur early and late in life. Events arrive at irregular intervals unlike in canonical clinical studies; see, *e.g.*, Figure 2.3.
- **Relational data.** To use most standard machine learning methods, data must be preprocessed into a flattened feature format which causes a loss of information and introduces statistical skew using autocorrelation and linkage (Jensen & Neville, 2002).

Pt ID	Date	Diagnosis/Prescription/Procedure
207a3d56	2007.7	Lipitor
207a3d56	2010.8	Chest pain
207a3d56	2010.83	Angina pectoris
207a3d56	2011.2	Myocardial infarction

Pt ID	Date	Laboratory Test	Laboratory Value
207a3d56	2007.7	Cholesterol	High
207a3d56	2007.7	LDL	High
207a3d56	2008.7	LDL	Normal
207a3d56	2010.83	LDL	Normal

Pt ID	Gender	Date of Birth
207a3d56	Male	1962.34

Pt ID	Date	Vital Type	Vital Value
207a3d56	2007.7	BP	High
207a3d56	2007.7	BMI	Overweight
207a3d56	2008.7	BP	Normal
207a3d56	2010.83	BP	High

Figure 2.1: Example of patient-specific tables in the EHR. The EHR database consists of tables including information such as diagnoses, drugs, labs, and genetic information.

- **Definition shifts.** Disease definitions are changing; subcategories and new types are introduced. New modalities in imaging and sequencing affect disease identification procedures and alter treatment guidelines. The medical trajectory of a patient one decade ago is different than it is today, making generalizations across time prone to bias.

While these challenges have been presented in the medical diagnosis framework, the nature of the data is not specific to this application. What the data capture are event timelines embedded in a relational domain. Algorithms for prediction in relational and continuous-time domains exist individually, but to our knowledge none exist that scalably and efficiently address this problem formulation. We develop methods that better handle these relational and temporal challenges in Chapter 3 through Chapter 6.

2.2 Statistical Relational Learning

To preface material in Chapter 3, we introduce Statistical Relational Learning (SRL). Relational models describe the relationships between objects, often using logic, which allows for more expressive descriptions than the classical alternative: an object as a vector in feature space. Many relational algorithms are extensions of their classical machine learning counterparts and are upgraded to the relational domain. SRL is the field that bridges relational modeling and probabilistic model learning. For example, relational probability trees are decision trees upgraded to first-order logic, and relational functional gradient boosting is the relational extension of functional gradient boosting (Neville et al., 2003; Natarajan et al., 2011b). Upgraded probabilistic models such as these comprise a major fraction of SRL methods. From the other perspective, established relational methods in databases and theorem-proving have been extended to corresponding probabilistic representations (Cavallo & Pittarelli, 1987; Raedt, 2008) and also fall within the field of SRL. All of these methods attempt to capture probabilistic behavior in richer, relational domains; see (Getoor & Taskar, 2007) for more examples.

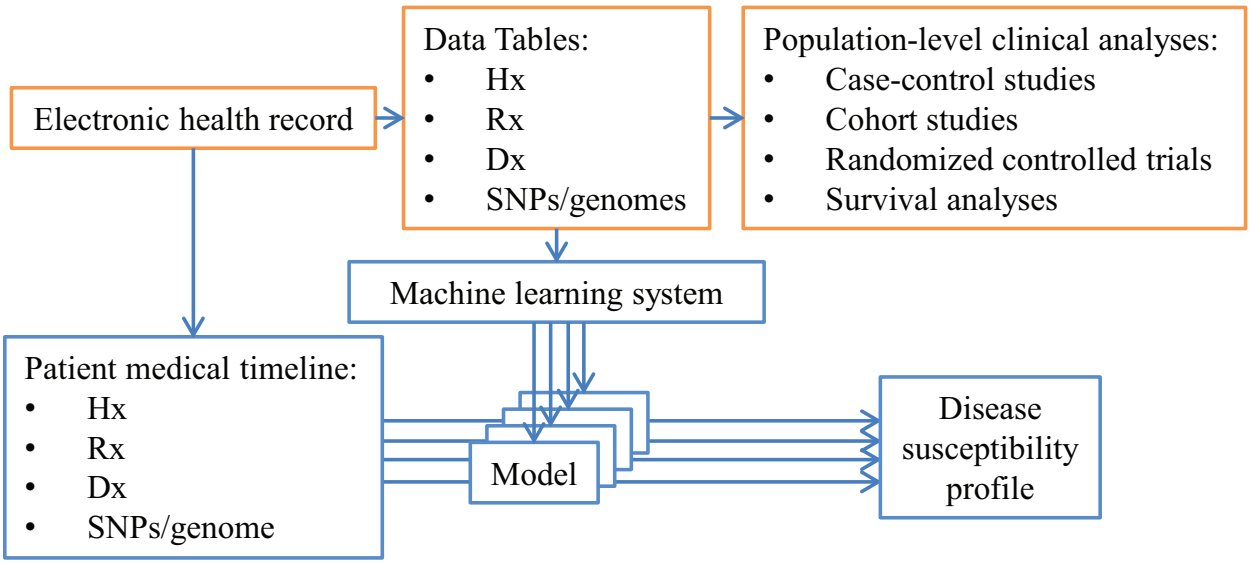

Figure 2.2: Machine learning systems (blue) can augment current clinical analyses (orange) by producing personalized health profiles given medical timelines of incoming patients. The clinical analyses typically identify and quantify risk factors that lead to disease; machine learning models integrate such risk factors into comprehensive predictive models. Medical history (Hx), drugs prescribed (Rx), and diagnoses (Dx) are abbreviated.

The primary advantage of SRL methods is their ability to work with the structure and relations in data; that is, information about one object helps the learning algorithms to reach conclusions about other objects. This helps in two primary ways.

- Examples are no longer assumed to be drawn i.i.d. from some underlying distribution, which is an impractical assumption in many domains. When relations between examples are provided in the data, e.g. if one subject is a sibling of another, SRL algorithms incorporate these relations and use them in their predictions.

- Complex objects can be better represented in the relational domain. In the medical prediction task, if patients have multiple blood pressure measurements, a relational framework can record each one, whereas a propositional framework requires either making aggregation design decisions or moving to a multiple instance problem setup.

Figure 2.4 shows a diagram illustrating the interconnectedness of pertinent health information for a medical diagnosis prediction problem. It depicts the relationships between patients, diagnoses, medications, and environment. It depicts the hierarchical nature of clinical records, for example having zip codes in states and diagnosis types in ICD-9 categories. Finally, it depicts an event as a set of objects at a particular time and place. In clinical studies we are often interested in the temporal health trajectory of a patient. The relationships in Figure 2.4, and many others that were omitted for clarity, form a complicated web of information. By using SRL algorithms we directly incorporate the structure of the domain, avoiding lossy feature extraction methods and modeling with fixed-length features.

The challenges associated with probabilistic relational algorithms typically center around the difficulty of scaling to large data sets. The three main machine learning tasks are defining model representation, doing model learning, and performing inference. Each task can be challenging in relational domains. For model

EHR data	Time	Framingham study measurements (FSM)	Framingham score dependencies
(FSM)	0	labs, physical exam (PE), medical history (Hx)	-cholesterol, +HDL, -blood pressure (-BP), smoker
+BP, hydrochlorothiazide -BP tachycardia (FSM)	2	labs, PE, Hx	-cholesterol, -HDL, -BP, smoker
+BP atrial fibrillation beta blocker, calcium channel blocker -BP (FSM)	4	labs, PE, Hx	-cholesterol, -HDL, -BP, smoker

Figure 2.3: A diagram comparing EHR data extracted to timelines (left) and Framingham Heart Study (FHS) data collection as a time series (right). The Framingham health cohort requires clinic visits every other year to perform laboratory assays (e.g. cholesterol levels), conduct physical exams including blood pressure measurements (BP), and document medical history (e.g. smoking status). The EHR contains FHS data and additional medical information with accurate timestamps, shown on the left. The Framingham Risk Score (FRS) is recalculated every two years, whereas one based on the EHR would be updated as new clinical events occur.

representation, a probability distribution needs to be defined over some space; common spaces include possible worlds (e.g. in Bayesian networks and probabilistic databases) or possible proofs (e.g. stochastic context free grammars). Model learning, split into structure and parameter learning, often requires expanding the relationship graph to the grounded network including all relationships among and within individual examples. The size of the ground network may be exponential in the number of examples or worse, making learning difficult. Model inference presents similar challenges in scalability, as the goal or query may require finding the distribution over the joint probability space encompassing the exponential-size ground network.

2.3 Continuous-Time Bayesian Networks

We turn to temporal analyses, where analyses over time series data with fixed, discrete time intervals predominate, as for example in Dean & Kanazawa (1989). However there are many domains in which discretizing the time leads to intervals where no observations are made, producing "missing data" in those periods, or there is no natural discretization available and so the time series assumptions are restrictive. Of note, experiments in previous work provide evidence that coercing continuous-time data into time series and conducting time series analysis is less effective than learning models built with continuous-time data in mind (Nodelman et al., 2003).

The prevailing model in continuous-time discrete state analysis is the continuous-time Markov process (CTMP), a model that provides an initial distribution over states and a rate matrix parameterizing the rate of transitioning between states. However, this model does not scale for the case where a CTMP state is a joint state over many variable states. Because the number of joint states is exponential in the number of variables, the size of the CTMP rate matrix grows exponentially in the number of variables. Continuous-time Bayesian networks (CTBNs), a family of CTMPs with a factored representation, encode rate matrices for each variable and the dependencies among variables (Nodelman, 2007). Figure 2.5 shows a complete trajectory,

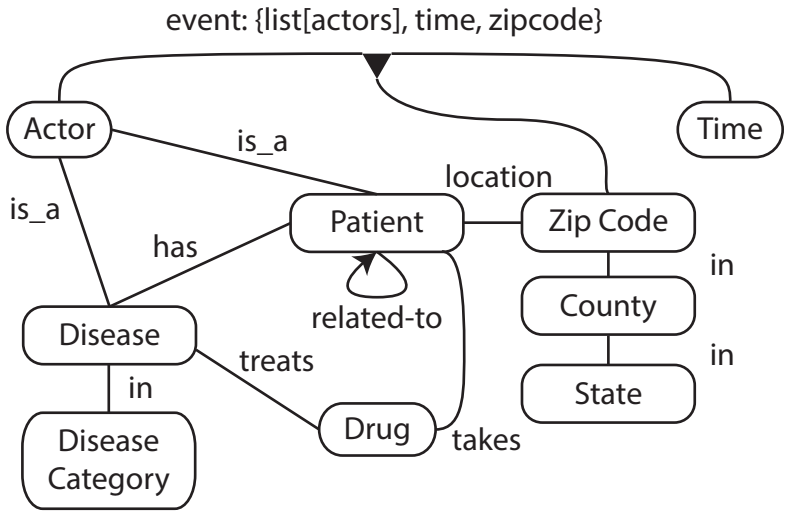

Figure 2.4: Schematic of the relationships one might be interested in modeling in the medical diagnosis domain. Each node represents an entity type, and the edges represent relationships among the entities that describe pertinent information about the domain. These complicated relationships challenge the notion of i.i.d data and fixed-size representations. In Ahmadi et al. (2012), the relational example is referred to as single mega-example, as each so-called example is intertwined with others due to the relationships among them.

i.e., a timeline where the state of each variable is known for all times t, for a CTMP with four joint states (a, b), (a, B), (A, b), and (A, B) factorized into two binary CTBN variables α and β (with states a and A, and b and B, respectively).

Formally, CTBNs are probabilistic graphical models that capture dependencies between variables over continuous time. A CTBN is defined by 1) a distribution for the initial state over variables $\mathcal{X}$ given by a Bayesian Network $\mathcal{B}$, and 2) a directed (possibly cyclic) graph over variables $\mathcal{X}$ with a set of *Conditional Intensity Matrices* (CIMs) for each variable $X \in \mathcal{X}$ that hold the rates (intensities) $q_{x|u}$ of variable transitions given their parents U_X in the directed graph. Here a CTBN variable $X \in \mathcal{X}$ has states $x^1, \ldots, x^k$, and there is an intensity $q_{x|u}$ for every state $x \in X$ given an instantiation over its parents $u \in U_X$. The intensity corresponds to the rate of transitioning out of state x; the probability density function for staying in state x given an instantiation of parents u is $q_{x|u}e^{-q_{x|u}t}$. Given a transition, X moves to some other state x' with probability $\Theta_{xx'|u}$.

The likelihood of a CTBN model given data is computed as follows. A trajectory is a sequence of intervals of fixed state. For each interval $[t_0, t_1)$, the duration $t = t_1 - t_0$ passes, and a variable X transitions at t_1 from state x to x'. During the interval all other variables $X_i \neq X$ remain in their current states x_i. The interval likelihood is given by:

$$\underbrace{q_{x|u}e^{-q_{x|u}t}}_{X \text{ transitions}} \underbrace{\Theta_{xx'|u}}_{\text{to state } x'} \underbrace{\prod_{x_i : X_i \neq X} e^{-q_{x_i|u}t}}_{\text{while } X_i\text{'s rest}}. \tag{2.1}$$

Taking the product over intervals bounded by single transitions, we obtain the CTBN trajectory likelihood:

$$\prod_{X \in \mathcal{X}} \prod_{x \in X} \prod_{u \in U_X} q_{x|u}^{M_{x|u}} e^{-q_{x|u}T_{x|u}} \prod_{x' \neq x} \Theta_{xx'|u}^{M_{xx'|u}} \tag{2.2}$$

where the $M_{x|u}$ and $M_{xx'|u}$ are the sufficient statistics indicating the number of transitions out of state x

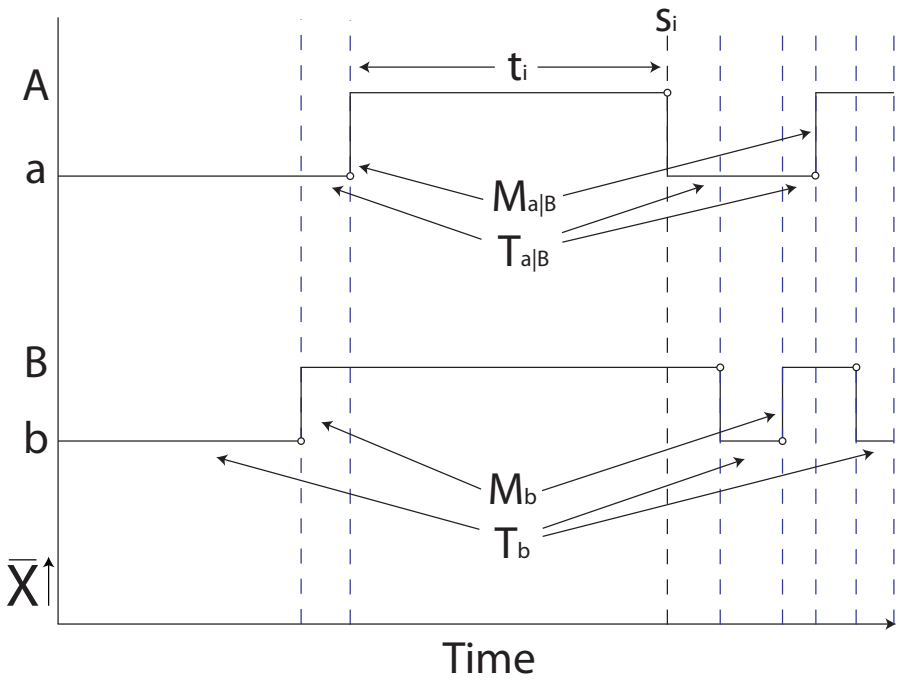

Figure 2.5: Example of a complete trajectory in a two-node CTBN. The arrows show the transitions and time intervals that are aggregated to compute selected sufficient statistics (M's and T's). A and a denote two states for one variable, and B and b two states for a second variable (left). The cardiovascular health (CV health) structure used in experiments (right).

(total, and to x', respectively), and the $T_{x|u}$ are the sufficient statistics for the amount of time spent in x given the parents are in state u.

The CTBN model provides a generative framework for forward sampling a trajectory z defined by a sequence of (state,time) pairs $z_i = (\{x_{1i}, x_{2i}, \ldots, x_{ni}\}, t_i)$, where x_{ji} is the jth CTBN variable at the ith time. Given an initial state $\{x_{10}, x_{20}, \ldots, x_{n0}\}$:

- Transition times are sampled for each variable x_j according to $q_{x_j|u}$.
- The one variable x_j that transitions is selected based on the sampled transition with the shortest time. The state that x_j transitions to is sampled from the multinomial $\theta_{x_j x'_j|u}$.
- The transition times are resampled according to intensities $q_{x_j|u}$, noting that these intensities may be different because of potential changes in the parents setting u. Due to the memoryless property of exponential distributions, no resampling of the transition time for x_j is needed if the intensity $q_{x_j|u}$ is unchanged.

The trajectory terminates when all sampled transition times exceed a specified ending time.

Figure 2.6 shows the graphical model representation of the first published CTBN network (Nodelman, 2007). Note that the graph is directed and contains a cycle. Cycles are allowed because the parents setting determines the child's rate of transitioning instead of the child's state. Thus factorization of the likelihood does not require the acyclicity constraint imposed in Bayesian networks. Similar to discrete state Bayesian

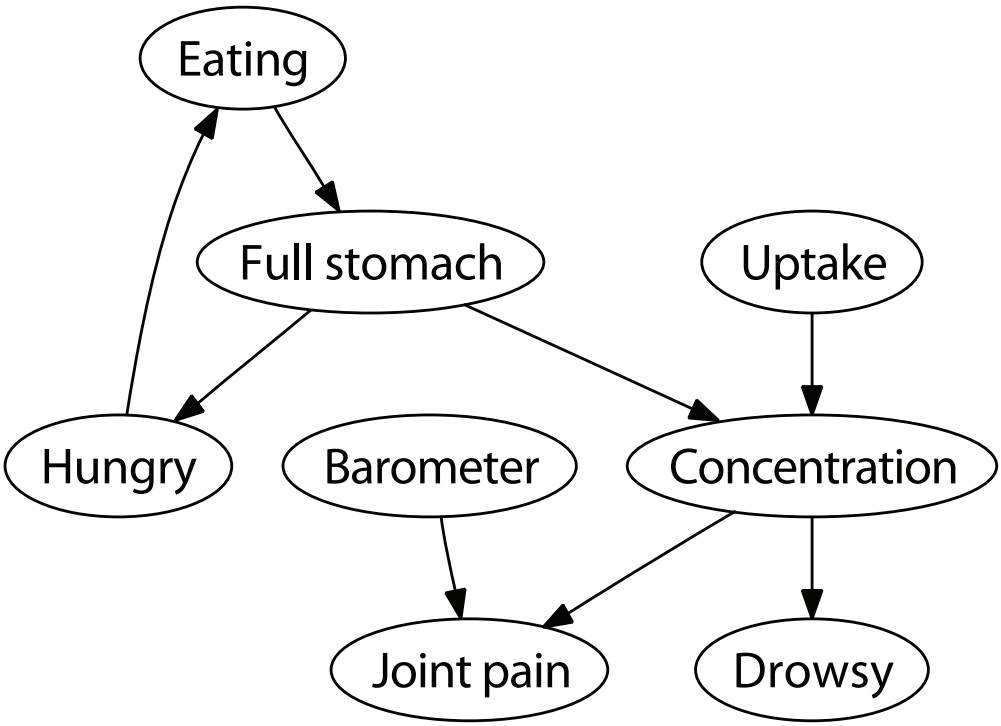

Figure 2.6: The Nodelman CTBN drug network. Note the graphical model representation for CTBNs allows cycles.

networks, the parameter space grows exponentially in the number of parents per variable. This limits the scalability of CTBNs; for example, in the model in Figure 2.6, the maximum number of incoming edges into a node is two. In line with context specific independence, our previous work has addressed how to maintain compact representations and facilitate efficient learning in such systems.

When CTBN trajectories have durations of time where the state of events are not completely observed, inference becomes necessary. Previous work on CTBN inference includes Nodelman et al. (2005); Saria et al. (2007); Cohn et al. (2009); Fan & Shelton (2008); Rao & Teh (2011), and we focus on extensions to the approximate inference methods (Fan & Shelton, 2008; Rao & Teh, 2011). Specifically, we seek to extend the sequential importance sampling methods presented in Fan & Shelton (2008). To build upon this work, we give a brief review of importance sampling and its sequential extension.

2.4 Sequential Importance Sampling

In this section we provide the basic problem setups for importance sampling and sequential importance sampling. These methods produce samples from generative models; in particular if we want to sample from a target distribution f, we can generate samples from surrogate distribution g, where each sample comes with a weight. The weighted distribution of samples from g takes into account our sampling of g so will approximate f if we generate enough samples.

Formally, let f be a p.d.f. defined on an ordered set of random variables $Z = \{Z_1, \ldots, Z_k\}$ over an event space Ω. We are interested in the conditional distribution $f(z|e)$, where evidence e is a set of observations about a subset κ of values $\{Z_i = z_i\}_{i \in \kappa}$. For fixed e, we define our target p.d.f. $f^*(z) = f(z|e)$. Let $g(z)$ be a surrogate distribution from which we can sample such that if $f^*(z) > 0$ then $g(z) > 0$. Then for any subset

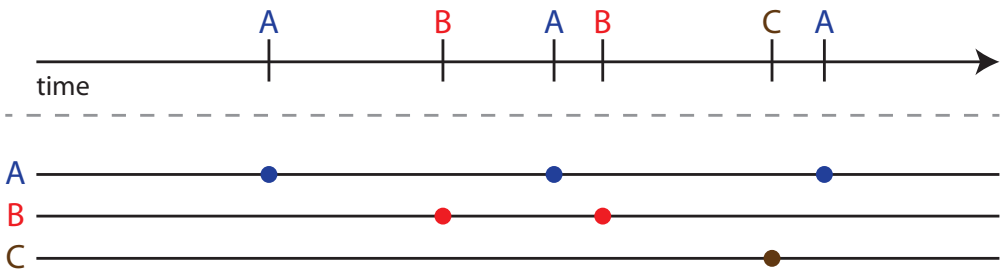

Figure 2.7: A timeline (top) deconstructed into point processes (bottom).

$\mathcal{Z} \subseteq \Omega$, we can approximate f^* with n weighted samples from g:

$$\int_{z \in \mathcal{Z}} f^*(z)dz = \int_{\mathcal{Z}} \frac{f^*(z)}{g(z)} g(z)dz \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[z^i \in \mathcal{Z}] \frac{f^*(z^i)}{g(z^i)} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[z^i \in \mathcal{Z}] w^i$$

where $\mathbb{1}[z^i \in \mathcal{Z}]$ is the indicator function with value 1 if $z^i \in \mathcal{Z}$ and 0 otherwise, and w^i is the importance sample weight.

Sequential importance sampling (SIS) is used when estimating the distribution f^* over the factorization of Z. In time-series models, Z_i is a random variable over the joint state corresponding to a time step; in continuous-time models, Z_i is the random variable corresponding to an interval. Defining $z_{j \leftarrow i} = \{z_j, z_{j-1}, \ldots, z_i\}$ for $i, j \in \{1, \ldots, k\}$ and $j \geq i$, we have the decomposition:

$$\begin{aligned} f^*(z) &= p(z_1, \ldots, z_k | e) \\ &= p(z_1|e) \prod_{i=2}^{k} p(z_i | z_{(i-1) \leftarrow 1}, e) \\ &= g_1(z_1|e) w_1(z_1|e) \prod_{i=2}^{k} g_i(z_i | z_{(i-1) \leftarrow 1}, e) w_i(z_i | z_{(i-1) \leftarrow 1}, e) \end{aligned} \tag{2.3}$$

where $p(\cdot)$ is the probability distribution under f. Equation 2.3 substitutes p with p.d.f. g_i by defining functions g_i and $w_i(\cdot) = p(\cdot)/g_i(\cdot)$ and requiring g_i to have the same support as p. Then g is defined by the composition of g_i: $g(z|e) = g_1(z_1|e) \prod_{i=2}^{k} g_i(z_i | z_{(i-1) \leftarrow 1}, e)$, and likewise for w. To generate a sample z^j from proposal distribution $g(z|e)$, SIS samples each z_i in order from 1 to k.

2.5 Point Processes

A complementary timeline formulation to CTBNs are point processes, which avoid CTBN inference challenges altogether. Instead of modeling variable states for every time t in some duration, point processes simply model the variable events that occur in the duration.

We can think of a timeline as a sequence of {event,time} pairs capturing the relative frequency and ordering of events and is a representation that arises in many domains, including neuron spike trains (Brown et al., 2004), high-frequency trading (Engle, 2000), and medical forecasting (Diggle & Rowlingson, 1994). A point process is a model that characterizes the distribution over emissions of an individual event over time. Thus, the point process treats each timeline event type individually and specifies that it (re-)occurs according to the intensity (or rate) function $\lambda(t|h)$ over time t given an event history h.

Figure 2.7 shows a sample timeline of events deconstructed into individual point processes. The conditional intensity model (CIM) is a probabilistic model formed by the composition of individual point processes.

Let us consider the finite set of event types $l \in \mathcal{L}$. An event sequence or trajectory x is an ordered set of {time, event} pairs $(t, l)_{i=1}^{n}$. A history h at time t is the subset of x whose times are less than t. Let l_0 denote the null event type, and use the null event pairs (l_0, t_0) and (l_0, t_{end}) to denote the start and end times of the trajectory. Then the likelihood of the trajectory given the CIM θ is:

$$p(x|\theta) = \prod_{l \in \mathcal{L}} \prod_{i=1}^{n} \lambda_l(t_i|h_i, \theta)^{\mathbb{1}(l=l_i)} e^{\int_{-\infty}^{t} \lambda_l(\tau|x,\theta)d\tau}$$

If we assume that $\lambda_l(t_i|h_i, \theta)$ is constant,

$$p(x|S, \theta) = \prod_{l \in \mathcal{L}} \lambda_l^{M_l(x)} e^{-\lambda_l T_l(x)} \tag{2.4}$$

where $M_l(x)$ is the count of events of type l in trajectory x, and $T_l(x)$ is the total duration l is modeled. In Chapter 6, we will leverage the similarity between Equations 2.2 and 2.4 to show that the learning frameworks we develop apply to each type of model.

2.6 Clinical Study Designs

Previous sections have described modeling frameworks to address temporal and relational aspects of EHR data. Here we provide background regarding the predominant use of clinical data–risk attribution–to give insight into the integration of EHR-based machine learning into existing biostatistics analyses.

The randomized controlled trial (RCT) is the primary risk attribution method. It randomizes patients to different treatment arms and measures the rate or probability of an outcome. The treatment arm with the highest success rate determines the preferred treatment, and the conclusion is that future patients who fit the entry criterion of the study should get the preferred treatment. Randomization is crucial to balance confounders, which are covariates that lead to the outcome and are associated with the treatment. Randomization also balances unmeasured confounders, so the study conclusion is free of confounding bias in expectation. The quantitative outcome of the RCT study is the average treatment effect (ATE), the average difference in probability of the outcome between two treatment arms.

In general, one cannot know what will happen to a specific patient under each treatment arm. The treatment that is given elicits the "true" outcome, and the treatment(s) not given elicits the "counterfactual" outcome. The counterfactual outcome is impossible to measure, but with randomization and the assumption that patients are drawn from an underlying population distribution, the expected outcome of patients assigned to a treatment arm is the same as the expected outcome of patients with the same treatment, true or counterfactual. Thus, RCTs provide a recommendation about the treatment effect for every treatment arm in the study for every patient.

The RCT is not feasible in many cases. Randomization to harmful treatments is unethical; for example, one does not randomize patients to "smoking" and "non-smoking" treatment arms. In such cases, observational studies are used to derive risk attribution statements, and these include cohort and case-control studies. Observational studies make the no unobserved confounders assumption (NUCA); the techniques rely upon modeling to pseudo-randomize the population distribution, but cannot do so effectively if they are missing

important contributors to their model–the unobserved confounders. Observational studies are designed to produce estimates for either the odds ratio, which can in turn be used to estimate the relative risk and the average treatment effect, or use pseudo-randomization techniques to mimic RCT data distributions.

When estimating the odds ratio, a conditional probability distribution (CPD) is used to model the probability of the outcome given the treatment and covariates as in, *e.g.*, Prentice (1976). The key is to include all confounders as covariates in the model, but not to include any intermediate variables. Intermediate variables are variables whose value are determined in part by the treatment and in turn affect the outcome, *i.e.*, they are on the "causal" pathway. Logistic regression models are often used and have the convenient characteristic that the coefficient associated with the treatment variable corresponds to the log odds ratio.

When using pseudo-randomization techniques, the idea is to re-weight the population distribution to make the treatment independent of covariates given the outcome. Propensity score matching constructs a model–typically a logistic regression–to stratify patients based on their propensity of treatment, matches patients within strata, and uses the matched population as its data set (Austin, 2011; Rosenbaum & Rubin, 1983). An alternative is to weight examples by the inverse probability-of-treatment (IPT); this involves modeling the IPT, weighting examples by 1 divided by the weights, and estimating the ATE from the pseudo-randomized population (Robins et al., 2000). A stabilized IPT weighting scheme is often used to reduce the potentially-large weight variance.

Combinations of these approaches exist: *e.g.*, the doubly-robust method using IPT and then modeling the CPD from the weighted distribution (Bang & Robins, 2005). The doubly-robust method is consistent if either the IPT estimator or the CPD model is properly specified. Unfortunately proper specification is often difficult to achieve and hard to assess in practice.

All of the above methods estimate the ATE, and there is a growing interest in modeling the individual treatment effect (ITE). The ITE is preferable because ITE-recommendations are patient- not population-specific. As a preview of Chapter 7, we suggest that machine learning could be useful in ITE estimation, and with developments in statistical timeline analysis, EHR data could become a leading source for future risk attribution findings.

3 Learning Relational Forests to Predict Primary Myocardial Infarction from Electronic Health Records

Overview

The previous chapter provided background important for Statistical Timeline Analysis. This chapter focuses on the relational challenges of the data. In particular, EHR data come from multiple tables potentially with different fields. This data representation cannot be practically coerced into fixed-length feature vectors, the primary data representation for machine learning and statistics, without losing information. To address these issues, relational learning uses the multiple table structure directly, and we adopt one such approach to leverage the relations available in EHR data. We apply two statistical relational learning (SRL) algorithms to the task of predicting primary myocardial infarction. We show that one SRL algorithm, relational functional gradient boosting, outperforms propositional learners particularly in the medically-relevant high recall region. We observe that both SRL algorithms predict outcomes better than their propositional analogs and suggest how our methods can augment current epidemiological practices. Similar versions of the work in this chapter were published in the Artificial Intelligence Magazine and Proceedings of the Innovative Applications of Artificial Intelligence (Weiss et al., 2012b;a).

3.1 Introduction

One of the most studied pathways in medicine is the health trajectory leading to heart attacks, known clinically as myocardial infarctions (MIs). MIs are common and deadly, causing one in three deaths overall in the United States totaling 600,000 per year (Manson et al., 1992). Because of its medical significance, MI has been studied in depth, mostly in the fields of epidemiology and biostatistics, yet rarely in machine learning. So far, it has been established that prediction of future MI is a challenging task. Risk stratification has been the predictive tool of choice (Diverse Populations Collaborative Group, 2002; Wilson et al., 1998), but these methods cannot reliably isolate the negative class; that is, everyone is still at risk. A much richer area of study is the identification of risk factors for MI. Common risk factors have been identified such as age, gender, blood pressure, low-density lipoprotein (LDL) cholesterol, diabetes, obesity, inactivity, alcohol and smoking. Studies have also identified less common risk factors as well as subgroups with particular risk profiles (Greenland et al., 2010; Antonopoulos, 2002).

The canonical method of study in this field is the identification or quantification of the risk attributable to a variable in isolation using: case-control studies, cohort studies, and randomized controlled trials. Case-control or cross-sectional studies identify odds ratios for the variable (or exposure) while controlling for confounders to estimate the relative risk. Cohort studies measure variables of interest at some early time point and follow the subjects to observe who succumbs to the disease. Randomized controlled trials are the gold standard for determining relative risks of single interventions on single outcomes. Each of these methods is highly focused, centered on the goal of providing the best risk assessment for one particular variable. One natural question to ask is: by using machine learning, can we conduct fewer studies by analyzing the effects of many variables instead?

A different and crucial limitation of the longitudinal methods is that they make measurements at fixed

points in time. In these studies, data is collected at the study onset t_0 to serve as the baseline variables, whose values are the ones used to determined risk. To illustrate this, consider the Skaraborg cohort study (Bog-Hansen et al., 2007) for the identification of acute MI mortality risk factors. The study measured established risk factors for MI at t_0, and then the subjects participated in annual checkups to assess patient health and determine if an MI event had occurred. It is important to note that, in line with current practice, the subjects who did not possess risk factors at time t_0 but developed them at some later time were considered as not possessing them in the analysis. If we knew that these developments had occurred, say from an EHR, would it be possible to estimate the attributable risk more precisely? In the extreme, can we estimate the risk factors and make reliable predictions without the annual checkups and the baseline t_0 measurements?

More generally, can we bring a machine learning perspective to this task that provides new insights to the study of MI prediction and risk factor identification? The answer is yes, and we present here a glimpse of the potential machine learning has to bring to this field. We suggest that the emergence of the EHR as the new data source for population health analyses may be able to answer these clinical questions more efficiently, effectively adding another method of study to the standard three. For the prediction task, we emphasize the evaluation of methods on statistics that are clinically relevant, specifically on class separability (for risk stratification) and precision at high recalls (for use as a screening tool). Class separability, which can be directly assessed using ROC curves, is a well-established tool for risk stratification (Diverse Populations Collaborative Group, 2002). Evaluating precision at high recalls assesses an algorithm's ability to predict while disallowing many false negatives, which is the critical component to a good screening tool. For predicting MI, a false negative means categorizing a patient as "low-risk" who goes on to have a heart attack, a costly outcome we wish to avoid. We also focus our methodology on algorithms with good interpretability, as this is critical for using the models for risk factor identification. In this work we survey a host of established machine learning algorithms for their performance on this task and select the most promising algorithm for further analysis. We attempt to answer some of these questions by providing an EHR-based framework for prediction and risk factor identification.

As mentioned in Chapter 2, EHR data presents significant challenges to current machine learning methodology. If we hope to augment traditional clinical study analyses, we must be able to effectively address these challenges. A few of them are: size, time-stamped data, relational data, and definition shifts over time.

We use Relational Functional Gradient Boosting (RFGB) because it addresses all but the last challenge, which is difficult for any algorithm to capture. Notably, it is one of the few relational methods capable of learning from large data sets. Moreover, RFGB can incorporate time by introducing temporal predicates like $\mathit{before}(A, B)$:-$A < B$. Also, unlike most other state-of-the-art SRL algorithms, RFGB allows us to learn structure and parameters simultaneously and grows the number of models as needed. Hence, we apply RFGB (Natarajan et al., 2010) and relational probability trees (RPTs) (Neville et al., 2003) to the task of predicting primary myocardial infarction (MI). Our goal is to establish that, even for large scale domains such as EHRs, that relational methods, and in particular RFBG and RPTs, can scale and outperform propositional variants.

This chapter makes a few key contributions: First, we address the challenging problem of predicting MI in real patients and identify ways in which machine learning can augment current methodologies in clinical studies. Second, we address this problem using recently-developed SRL techniques, adapt these algorithms to predicting MI and present the algorithms from the perspective of this task. Third, the task of MI prediction is introduced to the SRL community. To our knowledge, this is the first work to use SRL methods to predict MI in real patients. Fourth, we focus our analysis on interpretable RPT models, making it easy to discern the relationship between different risk factors and MI. Finally, our paper serves as a first step to bridge the gap

between SRL techniques and important, real-world medical problems.

3.2 Tree-Based Statistical Relational Learning

Statistical Relational Learning (SRL) (Getoor & Taskar, 2007), also known as relational probabilistic models, model structure and relations in data; that is, information about one object helps the learning algorithms to reach conclusions about other objects. Unfortunately, most SRL algorithms have difficulty scaling to large data sets. One efficient approach that yields good results from large data sets is the relational probability tree (Neville et al., 2003). The performance increase observed moving from propositional decision trees to forests is also seen in the relational domain (Anderson & Pfahringer, 2009; Natarajan et al., 2010). One method called functional gradient boosting (FGB) has achieved good performance in the propositional domain (Friedman, 2001). We apply it to the relational domain for our task: the prediction and risk stratification of MI from EHRs.

Relational Probability Trees

RPTs (Neville et al., 2003) were introduced for capturing conditional distributions in relational domains. These trees upgrade decision trees to the relational setting and have been demonstrated to build significantly smaller trees than other conditional models and obtain comparable performance. We use a version of RPTs that employs the TILDE relational regression (RRT) learner (Blockeel & Raedt, 1998) where we learn a regression tree to predict positive examples (in this case, patients with MI) and turn the regression values in the leaves into probabilities by exponentiating the regression value and normalizing them. Hence, the leaves of the RPTs are still the probability that a person has an MI given the other attributes. The key advantage of TILDE is that it can use conjunctions of predicates in the inner nodes as against a single test by the traditional RPT learner. This modification has been shown to have better performance than RPTs by others (Natarajan et al., 2010; Anderson & Pfahringer, 2009). In RRTs, the inner nodes (i.e., test nodes) are conjunctions of literals and each RRT can be viewed as defining several new feature combinations, one corresponding to each path from the root to a leaf. The resulting potential functions from all these different RRTs still have the form of a linear combination of features but the features can be quite complex (Gutmann & Kersting, 2006). We use weighted variance as the criterion to split on in the inner nodes. We augment the RRT learner with aggregation functions such as *count, max, average* that are used in the standard SRL literature (Getoor & Taskar, 2007) thus making it possible to learn complex features for a given target. These aggregators are pre-specified and the thresholds of the aggregators are automatically learned from the data. Continuous features such as *cholesterol* level, *ldl*, *bmi*, etc. are discretized into bins based on domain knowledge.

Relational Functional Gradient Boosting

Assume that the training examples are of the form $(\mathbf{x}_i, y_i)$ for $i = 1, ..., N$ and $y_i \in \{0, 1\}$ where $y = MI$ and $\mathbf{x}$ represents the set of all observations about the current patient i. The goal is to fit a model $P(y|\mathbf{x}) \propto e^{\psi(y,\mathbf{x})}$. The standard method of supervised learning is based on gradient-descent where the learning algorithm starts with initial parameters θ_0 and computes the gradient of the likelihood function. A more general approach is to train the potential functions based on Friedman's gradient-tree boosting algorithm where the potential functions are represented by sums of regression trees that are grown stage-wise (Friedman, 2001). More formally, functional gradient ascent starts with an initial potential ψ_0 and iteratively adds gradients Δ_i. Thus, after m iterations, the potential is given by $\psi_m = \psi_0 + \Delta_1 + ... + \Delta_m$. Here, Δ_m is the functional gradient at

episode m and is

$$\Delta_m = \eta_m \times E_{x,y}[\partial/\partial\psi_{m-1} log\, P(y|x;\psi_{m-1})] \tag{3.1}$$

where η_m is the learning rate. Dietterich et al.(Dietterich et al., 2004) suggested evaluating the gradient at every position in every training example and fitting a regression tree to these derived examples i.e., fit a regression tree h_m on the training examples $[(x_i, y_i), \Delta_m(y_i; x_i)]$. They point out that although the fitted function h_m is not exactly the same as the desired Δ_m, it will point in the same direction, assuming that there are enough training examples. So ascent in the direction of h_m will approximate the true functional gradient. The same idea has later been used to learn several relational models and policies (Natarajan et al., 2010; Sutton et al., 2000; Kersting & Driessens, 2008; Natarajan et al., 2011a; Gutmann & Kersting, 2006).

Let us denote the MI as y and it is binary valued (i.e., occurrence of MI). Let us denote all the other variables measured over the different years as $\mathbf{x}$. Hence, we are interested in learning $P(y|\mathbf{x})$ where $P(y|\mathbf{x}) = e^{\psi(y;\mathbf{x})}/\sum_y e^{\psi(y;\mathbf{x})}$. Note that in the functional gradient presented in Equation 3.1, the expectation $E_{x,y}[..]$ cannot be computed as the joint distribution $P(\mathbf{x}, \mathbf{y})$ is unknown. Hence, RFGB treats the data as a surrogate for the joint distribution.

Instead of computing the functional gradients over the potential function, they are instead computed for each training example i given as $(\mathbf{x_i}, y_i)$. Now this set of local gradients form a set of training examples for the gradient at stage m. Recall that the main idea in the gradient-tree boosting is to fit a regression-tree on the training examples at each gradient step. In this work, we replace the propositional regression trees with relational regression trees (Gutmann & Kersting, 2006; Natarajan et al., 2010; Kersting & Driessens, 2008).

The functional gradient with respect to $\psi(y_i = 1; \mathbf{x_i})$ of the likelihood for each example $(\mathbf{x}_i, y_i)$ can be shown to be:

$$\frac{\partial \log P(y_i; \mathbf{x_i})}{\partial \psi(y_i = 1; \mathbf{x_i})} = I(y_i = 1; \mathbf{x_i}) - P(y_i = 1; \mathbf{x_i}),$$

where I is the indicator function that is 1 if $y_i = 1$ and 0 otherwise. The expression is very similar to the one derived in Dietterich et al.(Dietterich et al., 2004). The key idea in this work is to represent the distribution over MI of a patient as a set of RRTs on the features. These trees are learned such that at each iteration the new set of RRTs aim to maximize the likelihood of the distributions with respect to ψ. Hence, when computing $P(MI(X)|\mathbf{f}(X))$ for a particular patient X, given the feature set $\mathbf{f}$, each branch in each tree is considered to determine the branches that are satisfied for that particular grounding (x) and their corresponding regression values are added to the potential ψ.

3.3 Experimental Methods

We analyzed de-identified EHR data on $18,386$ subjects enrolled in the Personalized Medicine Research Project (PMRP) at Marshfield Clinic (McCarty et al., 2005; 2008). The PMRP cohort is one of the largest population-based bio-banks in the United States and consists of individuals who are 18 years of age or older, who have consented to the study and provided DNA, plasma and serum samples along with access to their health information in the EHR. Most of the subjects in this cohort received most, if not all, of their medical care through the Marshfield Clinic integrated health care system.

Case definition

Within the PMRP cohort, 1153 cases were selected using the first International Classification of Diseases 9th revision (ICD9) code of 410.0 through 410.1. Cases were excluded if the incident diagnosis indicated

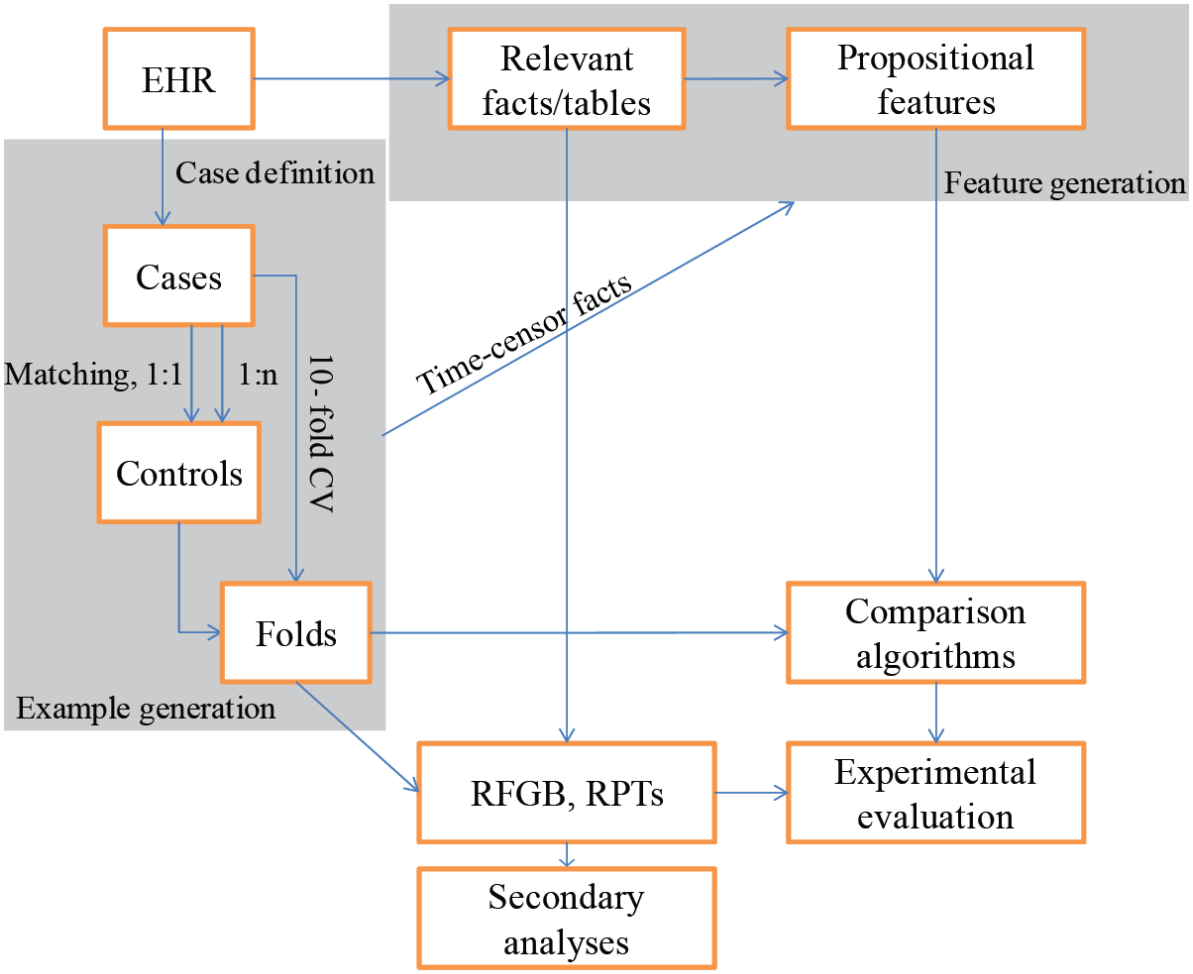

Figure 3.1: Flow chart depicting experimental setup

treatment for sequelae of MI or "MI with subsequent care". The age of the first case diagnosis was recorded and used to right-censor EHR data from both the case and the matching control one month prior to the case event. In other words, all facts linked to the case and the matched controls after the case age–one month prior to case diagnosis–were removed so that recent and future events could not be used in MI prediction.

Controls

To achieve a 1-1 ratio of cases to controls (i.e., positive and negative examples), cases were matched with controls based on the last age recorded in the EHR. For many matches, this corresponds to a case who is alive being matched to a control of the same age. For others it means matching someone who died from a heart attack to someone who died from other causes or was lost to follow-up. Matching on last reported age was chosen so that each subject would have both a similar age and similar presence in the EHR.

Feature selection

As CHD is the leading cause of mortality in the US, of which MI is a primary component, risk factors are well-studied (Antonopoulos, 2002; Greenland et al., 2010; Manson et al., 1992; Wilson et al., 1998), and those represented in the EHR were included in our experiments. We included major risk factors such as cholesterol levels (LDL in particular), gender, smoking status, and systolic blood pressure, as well as less common risk factors such as history of alcoholism and procedures for echocardiograms and valve replacements. Drugs known to have cardiac effects were included, notably the coxibs and tricyclic antidepressants. As EHR literals are coded in hierarchies, we chose to use the most specific level of information, which often split established risk factors into multiple subcategories. The risk factors were chosen a priori as opposed to employing algorithmic feature selection (e.g. the feature selection inherent in decision trees) to shrink the feature size from hundreds of thousands (excluding genetic data) to thousands for computational reasons and so that algorithms without inherent feature selection would perform comparably. The features chosen came from relational tables for diagnoses, medications, labs, procedures, vitals, and demographics.

Propositionalization

Patient relations were extracted to temporally-defined features in the form of "patient ever had $x \in X$" or "patient had $x \in X$ within the last year". For laboratory values and vitals, both of which require an additional literal for the result of the test, the result was binned into established value categories (e.g. for blood pressure, we created five binary features by mapping the real value to {critically high, high, normal, low, and critically low}). This resulted in a total of 1,528 binary features.

Evaluation measures

The cases and controls were split into ten folds for cross-validation in a nine-fold train set to one-fold test set. Although we did choose a one-to-one ratio of cases to controls, in general this would not be the case, so we chose to assess the performance of the algorithms with the area under the ROC curve (AUC-ROC), accuracy, and by visualizing the results with a precision-recall plot. Also, precision at high recalls {0.95, 0.99, 0.995} were calculated to assess a model's usefulness as a screening tool. p-values were calculated comparing the RFGB model with the comparison methods using a two-sided paired t-test on the ten-fold test sets, testing for significant differences in accuracy and precision at a recall of 0.99.

Comparison methods

The key question is whether the relational algorithms consistently produced better predictions than their corresponding propositional variant. Thus we compared RFGB models to boosted decision trees (AdaBoostM1 (Ada); default parameters) and RPTs with decision tree learners (J48; C=0.25, M=2). We also included other common models: Naive Bayes (NB; default parameters), Tree-Augmented Naive Bayes (TAN; SimpleEstimator), support vector machines (SVMs; linear kernel, C 1.0; radial basis function kernel, C 250007, G 0.01), and random forests (RF; 10 trees, default parameters). All propositional learners were run using Weka software (Hall et al., 2009).

Secondary analysis

In our secondary analysis, we varied both the experimental setup and the RFGB parameters to investigate the effect on their predictive ability. First, we altered the case-control ratio {1:1, 1:2, 1:3}, holding the number

Table 3.1: Area under the ROC curve, accuracy and corresponding p-value(RFGB vs. all), precision at recall (P@R), and p-value(RFGB vs. all, P@R=0.99). Bold indicates best performance.

	AUC-ROC	Accuracy	p	P@R=0.99	p(P@R=0.99)
Tree J48	0.744	0.716	4e-5	0.500	6e-7
Boosted Trees	0.807	0.753	1e-4	0.572	4e-4
Random Forests	0.826	0.785	4e-1	0.593	2e-3
NB	0.840	0.788	8e-1	0.513	1e-4
TAN	0.830	0.768	6e-3	0.518	2e-4
SVM (linear)	0.704	0.704	5e-6	–	–
SVM (rbf)	0.761	0.761	1e-2	–	–
RFGB	**0.845**	**0.791**	–	**0.655**	–
RPT	0.792	0.738	4e-6	0.595	4e-5

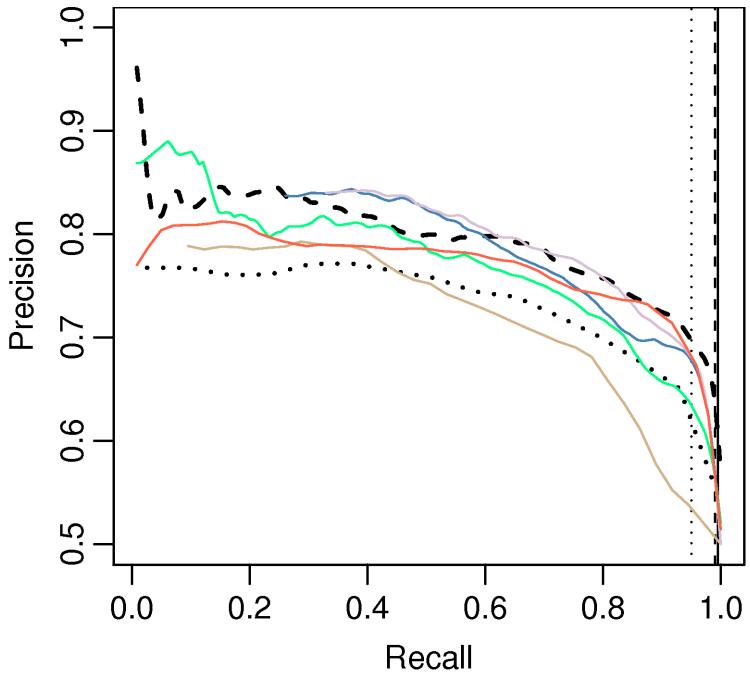

Figure 3.2: Precision-recall curves, with vertical lines denoting the recall thresholds {0.95, 0.99, 0.995}. RFGB (dashed) and RPT (dotted) are bolded. RFGB outperforms all other algorithms in the medically-relevant region (high recall). At recall=0.9, the ordering of algorithms (best to worst) is: RFGB, Random Forests, TAN, NB, RPT, Boosted Trees, J48.

of cases fixed. Second, we altered the maximum number of clauses (for internal node splits) allowed per tree {3, 10 (default), 20, 30}. Third, we altered the maximum depth of the tree {1 (stump), 5}. Finally, we altered the number of trees {3, 10 (default), 20, 30}. We also compared the results among these analyses if they contained the same maximum number of parameters (e.g. 30 parameters: 3 trees $\times$ 10 clauses, 10 trees $\times$ 3 clauses).

3.4 Results

The best cross-validated predictor of primary MI according to AUC-ROC was the RFGB model as shown in Table 3.1. RFGB outperformed the other tree learners, forest learners and SVMs. The RPT model did not

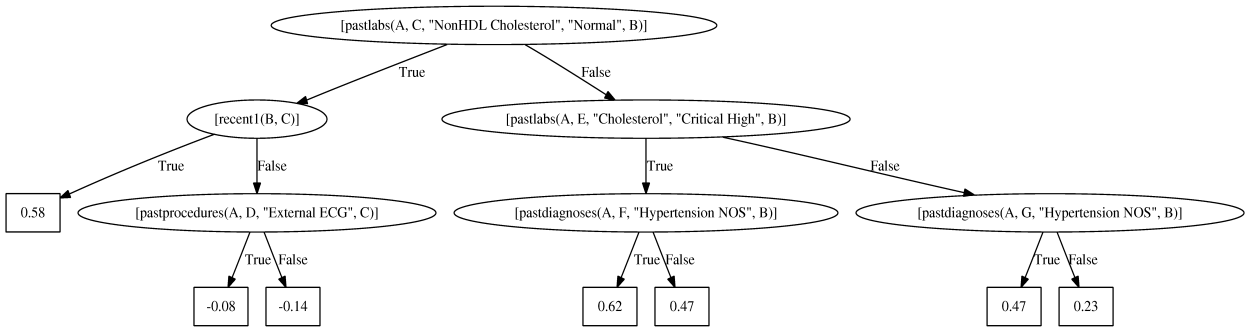

Figure 3.3: The first learned tree in the RFGB forest

score as well, ranking in the middle of the propositional learners. It is of note that the RFGB and RPT models significantly outperformed their direct propositional analogs (Boosted Tree and Tree models, respectively). The Bayesian model (NB; TAN) scores may be somewhat inflated because only features known to be CHD risk factors were specifically chosen for this analysis. They may be more prone to irrelevant feature noise as those models include all features into their final models.

The precision-recall curves for the algorithms are shown in Figure 3.2 (SVMs are omitted as their outputs do not admit a ranking over examples). Medically, the most important area is the region of high recall (i.e. sensitivity) because typically the cost of leaving a condition undiagnosed is high. In other words, the expected cost of a false positive is much smaller than a false negative because a false positive incurs the costs of additional interventions, while a false negative incurs costs of untreated human morbidity, and usually expensive, delayed treatments. Given that we cannot accept models with many false negatives (i.e. low recall), we look to the high recall region for the best performing algorithm, and RFGB gives the highest precision as shown in Table 3.1.

In our secondary analysis, when changing the case-control ratio we observed an increase in the AUC-ROC as well as the expected increase in accuracy and decrease in precision shown in Table 3.2. We suspect the improvement in AUC-ROC may be attributed to the larger population size, as for example CC 1:3 has twice as many examples as CC 1:1. RFGB performance improved with increases with forest size, with the greatest gains coming between using three and ten trees, and no overfitting was observed using our largest fifty-tree forest (see our website: `http://cs.wisc.edu/~jcweiss/iaai2012`). Varying the number of clauses or tree depth made no visible difference in RFGB performance, at least when holding the number of trees fixed at ten. Per parameter, we found that increasing forest size improved prediction more than increasing individual tree sizes, as we see by comparing equal-parameter rows in Table 3.2.

Figure 3.3 shows an example tree produced in the RFGB forest. We can read this as follows. Given a patient A and their censor age B (i.e. for cases, one month before their first MI; for controls, the censor age

Table 3.2: Secondary analyses: RFGB performance as case-control ratio (CC), number of clauses, trees and tree depth are modified. Default number of clauses = 10 and trees = 10

	AUC-ROC	Accuracy	P@R=0.99
CC 1:1;1:2;1:3	.84;.87;.88	.79;.80;.82	.66;.51;.43
Trees 3;20;30	.80;.85;.85	.74;.80;.80	.61;.67;.66
Clauses 3;20;30	.85;.85;.85	.79;.79;.79	.66;.66;.66
Tree depth 1;5	.85;.85	.79;.79	.66;.66

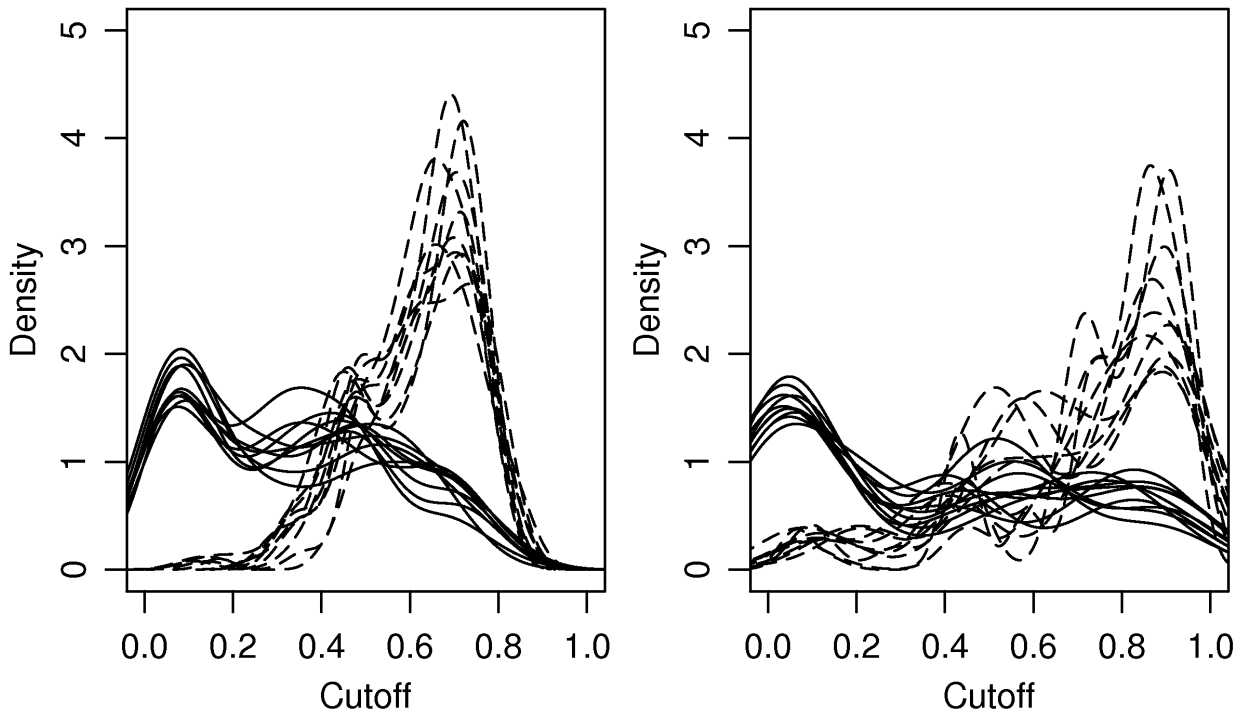

Figure 3.4: Density of cases (dashed) and controls (solid) by {RFGB (left), RPT (right)} prediction, one line per fold. Taking the integral from 0 to cutoff c for example at $c = 0.05$ and $c = 0.25$ shows that RFGB identifies many controls at low-risk of developing MI.

of the corresponding case), if A had a normal non-HDL cholesterol measurement at time C, take the left branch, otherwise take the right branch. Assuming we took the left branch, if the measurement C was within one year of the censor age, take the left branch again. The leaf regression value is the best estimate of the residual of the probability of the covered examples given the model at that iteration. The whole RFGB forest is available at our website: `http://cs.wisc.edu/~jcweiss/iaai2012`.

Direct interpretation of the tree can lead to useful insights. In the example above, the tree indicates that a patient is more likely to have a future MI event if they have had a normal non-HDL cholesterol level reading in the last year compared to patients who have had normal cholesterol readings not in the last year. Now, since it is implausible that the measurement itself is causing MI, it could be considered a proxy for another "risk factor", which in this case could be physician concern, as frequent lipoprotein measurements may display a concern for atherosclerosis-related illness. The set of trees can also be converted into a list of weighted rules to make them more interpretable (Craven & Shavlik, 1996).

The density plot in Figure 3.4 shows the ability of RFGB and RPT models to separate the MI class from the controls. It is clear from the far left region of the RFGB graph that we can accurately identify a substantial fraction of controls with few cases by thresholding around 0.25, or more stringently at 0.05. This region captures an algorithm's utility as a screening tool, where we see that RFGB significantly outperforms the others.

3.5 Discussion

One layer of complexity not addressed in this experiment is the use of other relational information such as hierarchies. EHRs have hierarchies for diagnoses, drugs, and laboratory values, and it is important to be able to capture detail at each level. For example, characteristic disease progression pathways stem from infarctions of different heart walls, but at a high level, the presence of any MI leads to standard sequelae. Relational domains can easily incorporate this knowledge into hierarchical "is a" relations, whereas propositional learners must create new features for every level. The challenge for relational tree-based learners is that the search algorithm is greedy; identifying high-level relations requires traversing several "is a" relationships first, and thus they might not be found in a greedy search. Expanding internal nodes to longer clauses has been implemented with some success (Natarajan et al., 2010; Anderson & Pfahringer, 2009), although this does have the effect of rapidly increasing the number of features to consider during branching. The use of SRL algorithms could also allow the use of relations like patient physicians and providers, which form complex relations less "patient-disease"-oriented but ones that still may be central to patient care. Questions regarding disease heritability could also be addressed through relational family-based analyses.

Given the success of the RFGB method, one extension would include the addition of more potential risk factors for learning (i.e., include all the measurements on all the patients). This could be challenging as the number and frequencies of the measurements differ greatly across patients. In the experimental RFGB model, we used time as the last argument of our predicates. While a vast body of work discusses learning and reasoning with temporal models in propositional domains, the situation is not the same for relational models. The investigation of a principled approach to learn and reason with relational dynamic models that allows physicians to monitor the cardiovascular risk levels of patients over time and develop personalized treatment plans could be extremely valuable. Finally, deployment of a complete machine learning system for identifying risk factors across many diseases given EHR data could immediately augment the clinical work flow.

3.6 Summary

In this chapter, we presented the challenging and high-impact problem of primary MI from an EHR database using a subset of known risk factors. We adapted two SRL algorithms in this prediction problem and compared them with standard machine learning techniques. We demonstrated that RFGB is as good as or better than propositional learners at the task of predicting primary MI from EHR data. Each relational learner does better than its corresponding propositional variant, and in the medically-relevant, high recall region of the precision-recall curve, RFGB outperforms all the other methods that were considered. One limitation of this method is that time could only be used logically in the forest learning algorithm. In the next three chapters, we explore explicit timeline models, which capture time in continuous fashion and can model over multiple time scales.

4 Learning Multiplicative Forests for Continuous-Time Bayesian Networks

Overview

We turn now to modeling EHR patient data as timelines in a continuous-time framework. This framework allows us to effectively learn temporal dependencies between variables at varying time scales, which is important medically because events tend to arrive in clusters. We adopt the continuous-time Bayesian network (CTBN) model, which effectively model events over continuous time. However, it is limited by the number of conditional intensity matrices, which grows exponentially in the number of parents per variable. We develop a partition-based representation using regression trees and forests whose parameter spaces grow linearly in the number of node splits. Using a multiplicative assumption we show how to update the forest likelihood in closed form, producing efficient model updates. Our results show multiplicative forests can be learned from few temporal trajectories with large gains in performance and scalability. A similar version of this chapter was published in the Proceedings of the Neural Information Processing Systems Conference (Weiss et al., 2012c).

4.1 Introduction

The modeling of temporal dependencies is an important and challenging task with applications in fields that use forecasting or retrospective analysis, such as finance, biomedicine, and anomaly detection. Many studies have analyzed temporal data using fixed, discrete time intervals, *e.g.*, Dean & Kanazawa (1989), but for many timelines, there is no natural discretization available making the time series assumption overly restrictive. Previous work provides evidence that using time series analysis on continuous-time data is less effective than using continuous-time models directly (Nodelman et al., 2003).

We specifically investigate probabilistic models over finite event spaces across continuous time, *i.e.*, continuous-time Markov process (CTMP). This model provides an initial distribution over states and a rate matrix parameterizing the rate of transitioning between states. However, it does not scale to joint states over many variable states because the number of joint states is exponential in the number of variables, and thus the size of the CTMP rate matrix grows exponentially in the number of variables. Continuous-time Bayesian networks (CTBNs) are a family of CTMPs with a factored representation that encode rate matrices for each variable and the dependencies among variables (Nodelman, 2007). Figure 2.5 shows an example of a trajectory, i.e., a timeline where the state of each variable is known for all times t, for a CTMP with four joint states (a, b), (a, B), (A, b), and (A, B) factorized into two binary CTBN variables α and β (with states a and A, and b and B, respectively).

Previous work on CTBNs includes several approaches to performing CTBN inference (Nodelman et al., 2005; Saria et al., 2007; Cohn et al., 2009; Fan & Shelton, 2008; Rao & Teh, 2011) and learning (Nodelman et al., 2003; Nodelman, 2007). Briefly, CTBNs do not admit exact inference without transformation to the exponential-size CTMP. Approximate inference methods including expectation propagation (Nodelman et al., 2005), mean field (Cohn et al., 2009), importance sampling-based methods (Fan & Shelton, 2008), and MCMC (Rao & Teh, 2011) have been applied, and while these methods have helped mitigate the inference

problem, inference in large networks remains a challenge. CTBN learning involves parameter learning using sufficient statistics (e.g. numbers of transitions M and durations T in Figure 2.5) and structure learning over a directed (possibly cyclic) graph over the variables to maximize a penalized likelihood score. Our work addresses learning in a generalized framework to which the inference methods mentioned above can be extended.

In this work we introduce a generalization of CTBNs: partition-based CTBNs. Partition-based CTBNs remove the restriction used in CTBNs of storing one rate matrix per parents setting for every variable. Instead partition-based CTBNs define partitions over the joint state space and define the transition rate of each variable to be dependent on the membership of the current joint state to an element (part) of a partition. As an example, suppose we have partition P composed of parts $p_1 = \{(a, b), (A, b)\}$ and $p_2 = \{(a, B), (A, B)\}$. Then the transition into s_i from joint state (A, B) in Figure 2.5 would be parameterized by transition rate $q_{a|p_2}$. Partition-based CTBNs store one transition rate per part, as opposed to one transition rate matrix per parents setting. Later we will show that, for a particular choice of partitions, a partition-based CTBN is equivalent to a CTBN. However, the more general framework offers other choices of partitions which may be more suitable for learning from data.

Partition-based CTBNs avoid one limitation of CTBNs: that the model size is necessarily exponential in the maximum number of parents per variable. For networks with sparse incoming connections, this issue is not apparent. However, in many real domains, a variable's transition rate may be a function of many variables.

Given the framework of partition-based CTBNs, we need to provide a way to determine useful partitions. Thus, we introduce partition-based CTBN learning using regression tree modifications in place of CTBN learning using graph operators of adding, reversing, and deleting edges. In the spirit of context-specific independence (Heckerman, 1993), we can view tree learning as a method for learning compact partition-based dependencies. However, tree learning induces recursive subpartitions, which limits their ability to partition the joint state space. We therefore introduce multiplicative forests for CTBNs, which allow the model to represent up to an exponential number of transition rates with parameters still linear in the number of splits.

Following canonical tree learning methods, we perform greedy tree and forest learning using iterative structure modifications. We show that the partition-based change in log likelihood can be calculated efficiently in closed form using a multiplicative assumption. We also show that using multiplicative forests, we can efficiently calculate the ML parameters. Thus, we can calculate the maximum change in log likelihood for a forest modification proposal, which gives us the best iterative update to the forest model.

Finally, we conduct experiments to compare CTBNs, regression tree CTBNs (treeCTBNs) and multiplicative forest CTBNs (mfCTBNs) on three data sets. Our hypothesis is twofold: first, that learning treeCTBNs and mfCTBNs will scale better towards large domains because of their compact model structures, and second, that mfCTBNs will outperform both CTBNs and treeCTBNs with fewer data points because of their ability to capture multiplicative dependencies.

The rest of the chapter is organized as follows: in Section 4.2 we provide background on CTBNs. In Section 4.3 we present partition-based CTBNs, show that they subsume CTBNs and define the partitions that tree and forest structures induce. We also describe theoretical advantages of using forests for learning and how to learn these models efficiently. We present results in Section 4.4 showing that forest CTBNs are scalable to large state spaces and learn better than CTBNs, from fewer examples and in less time. Finally, in Sections 4.5 and 4.6 we identify connections to functional gradient boosting and related continuous-time processes and discuss how our work addresses one limitation that prevents CTBNs from finding widespread use.

4.2 Background

CTBNs are probabilistic graphical models that capture dependencies between variables over continuous time. As a reminder, and for notational consistency, we reintroduce CTBNs here. Recall that a CTBN is defined by 1) a distribution for the initial state over variables $\mathcal{X}$ given by a Bayesian Network $\mathcal{B}$, and 2) a directed (possibly cyclic) graph over variables $\mathcal{X}$ with a set of *Conditional Intensity Matrices* (CIMs) for each variable $X \in \mathcal{X}$ that hold the rates (intensities) $q_{x|u}$ of variable transitions given their parents U_X in the directed graph. A CTBN variable $X \in \mathcal{X}$ has states $x^1, \ldots, x^k$, and there is an intensity $q_{x|u}$ for every state $x \in X$ given an instantiation over its parents $u \in U_X$. The intensity is the rate of transitioning out of state x; the probability density function for staying in state x given an instantiation of parents u is $q_{x|u}e^{-q_{x|u}t}$. Given a transition, X moves to some other state x' with probability $\Theta_{xx'|u}$. Taking the product over intervals bounded by single transitions, we obtain the CTBN trajectory likelihood:

$$\prod_{X \in \mathcal{X}} \prod_{x \in X} \prod_{u \in U_X} q_{x|u}^{M_{x|u}} e^{-q_{x|u}T_{x|u}} \prod_{x' \neq x} \Theta_{xx'|u}^{M_{xx'|u}}$$

where the $M_{x|u}$ and $M_{xx'|u}$ are the sufficient statistics indicating the number of transitions out of state x (total, and to x', respectively), and the $T_{x|u}$ are the sufficient statistics for the amount of time spent in x given the parents are in state u.

4.3 Partition-based CTBNs

Here we define partition-based CTBNs, an alternative framework for determining variable transition rates. We give the syntax and semantics of our model, providing the generative model and likelihood formulation. We then show that CTBNs are one instance in our framework. Next, we introduce regression trees and multiplicative forests and describe the partitions they induce, which are then used in the partition-based CTBN framework. Finally, we discuss the advantages of using trees and forests in terms of learning compact models efficiently.

Let $\mathcal{X}$ be a finite set of discrete variables X of size n, with each variable X having a discrete set of states $\{x^1, x^2, \ldots, x^k\}$, where k may differ for each variable. We define a joint state $s = \{x_1, x_2, \ldots, x_n\}$ over $\mathcal{X}$ where the subscript indicates the variable index. We also define the partition space $\mathcal{P} = \mathcal{X}$[1]. We will shortly define set partitions P over $\mathcal{P}$, composed of disjoint parts p, each of which holds a set of elements s.

Next we define the dynamics of the model, which form a continuous-time process over $\mathcal{X}$. Each variable X transitions among its states with rate parameter $q_{x'|s}$ for entering state x' given the joint state s[2]. This rate parameter (called an intensity) parameterizes the exponential distribution for transitioning into x', given by the pdf: $p(x', s, t) = q_{x'|s}e^{-q_{x'|s}t}$ for time $t \in [0, \infty)$.

A partition-based CTBN has a collection of set partitions P over $\mathcal{P}$, one $P_{x'}$ for every variable state x'. For shorthand, we will often denote $p = P_{x'}(s)$ to indicate the part p of partition $P_{x'}$ to which state s belongs. We define the intensity parameter as $q_{x'|s} = q_{x'|p}$ for all $s \in p$. Note that this fixes this intensity to be the same for every $s \in p$, and also note that the set of parts p covers $\mathcal{P}$. The pdf for transitioning is given by $p(x', s, t) = p(x', P_{x'}(s), t) = q_{x'|p}e^{-q_{x'|p}t}$ for all s in p.

[1]Note we can generalize this to larger spaces $\mathcal{P} = \mathcal{R} \times \mathcal{X}$, where $\mathcal{R}$ is an external state space as in (Gunawardana et al., 2011). but for our analysis we restrict $\mathcal{R}$ to be a single element r, i.e. $\mathcal{P} \cong \mathcal{X}$.

[2]Of note, partition-based CTBNs are modeling the intensity of transitioning to the recipient state x', rather than from the donor state x because we are more often interested in the causes of *entering* a state.

Now we are ready to define the partition-based CTBN model. A partition-based CTBN model $\mathcal{M}$ is composed of a distribution over the initial state of our variables, defined by a Bayesian network $\mathcal{B}$, and a set of partitions $P_{x'}$ for every variable state x' with corresponding sets of intensities $q_{x'|p}$.

The partition-based CTBN provides a generative framework for producing a trajectory z defined by a sequence of (state, time) pairs (s_i, t_i). Given an initial state s_0, transition times are sampled for each variable state x' according to $p(x', P_{x'}(s_0), t)$. The next state is selected based on the transition to the x' with the shortest time, after which the transition times are resampled according to $p(x', s_i, t)$. Due to the memoryless property of exponential distributions, no resampling of the transition time for x' is needed if $p(x', s_i, t) = p(x', s_{i-1}, t)$. The trajectory terminates when all sampled transition times exceed a specified ending time.

Given a trajectory z, we can also define the model likelihood. For each interval t_i, the joint state remains unchanged, and then one variable transitions into x'. The likelihood given the interval is: $q_{x'|s_{i-1}} \prod_X \prod_{x \in X} e^{-q_{x|s_{i-1}} t_i}$, i.e., the product of the probability density for x' and the probability that no other variable transitions before t_i. Taking the product over all intervals in z, we get the model likelihood:

$$\prod_{X \in \mathcal{X}} \prod_{x' \in X} \prod_{s} q_{x'|s}^{M_{x'|s}} e^{-q_{x'|s} T_s} \tag{4.1}$$

where $M_{x'|s}$ is the number of transitions into x' from state s, and T_s is the total duration spent in s. Combining terms based on the membership of s to p and defining $M_{x'|p} = \sum_{s \in p} M_{x'|s}$ and $T_p = \sum_{s \in p} T_s$, we get:

$$\text{Eq.}(4.1) = \prod_{X \in \mathcal{X}} \prod_{x' \in X} \prod_{p \in P_{x'}} q_{x'|p}^{M_{x'|p}} e^{-q_{x'|p} T_p}$$

CTBN as a partition-based CTBN

Here we show that CTBNs can be viewed as an instance of partition-based CTBNs. Each variable X is given a parent set U_X, and the transition intensities $q_{x|u}$ are recorded for *leaving* donor states x given the current setting of the parents $u \in U_X$. The CTBN likelihood can be shown to be:

$$\prod_{X \in \mathcal{X}} \prod_{x \in X} \prod_{u \in U_X} e^{-q_{x|u} T_{x|u}} \prod_{x' \neq x} q_{xx'|u}^{M_{xx'|u}} \tag{4.2}$$

as in (Saria et al., 2007), where $q_{xx'|u}$ and $M_{xx'|u}$ denote the intensity and number of transitions from state x to state x' given parents setting u, and $\sum_{x' \neq x} q_{xx'|u} = q_{x|u}$. Rearranging the product from equation 4.2, we achieve a likelihood in terms of recipient states x':

$$\begin{aligned} \text{Eq. (4.2)} &= \prod_{X \in \mathcal{X}} \prod_{x \in X} \prod_{u \in U_X} \prod_{x' \neq x} q_{xx'|u}^{M_{xx'|u}} e^{-q_{xx'|u} T_{x|u}} \\ &= \prod_{X \in \mathcal{X}} \prod_{x' \in X} \prod_{p \in P_{x'}} q_{x'|p}^{M_{x'|p}} e^{-q_{x'|p} T_p} \end{aligned} \tag{4.3}$$

where we define p as $\{x\} \times \{u\} \times (\mathcal{X} \setminus (X \times U_X))$ in each partition $P_{x'}$, and likewise: $q_{x'|p} = q_{xx'|u}$, $M_{x'|p} = M_{xx'|u}$, and $T_p = T_{x|u}$. Thus, CTBNs are one instance of partition-based CTBNs, with partitions corresponding to a specified donor state x and parents setting u.

Tree and forest partitions

Trees and forests induce partitions over a space defined by the set of possible split criteria (Strobl et al., 2009). Here we will define the Conditional Intensity Trees (CITs): regression trees that determine the intensities $q_{x'|p}$ by inducing a partition over $\mathcal{P}$. Similarly, we will define Conditional Intensity Forests (CIFs), where tree intensities are named intensity factors whose product determines $q_{x'|p}$. An example of a CIF, composed of a collection of CITs, is shown later in the experiment results in Figure 4.3.

Formally, a *Conditional Intensity Tree* (CIT) $f_{x'}$ is a directed tree structure on a graph $G(V, E)$ with nodes V and edges $E(V_i, V_j)$. Internal nodes V_i of the tree hold splits $\sigma_{V_i} = (\pi_{V_i}, \{E(V_i, \cdot)\})$ composed of surjective maps $\pi_{V_i} : s \mapsto E(V_i, V_j)$ and lists of the outgoing edges. The maps π induce partitions over $\mathcal{P}$ and endow each outgoing edge $E(V_i, V_j)$ with part p_{V_j}. External nodes l, or leaves, hold non-negative real values $q^{\text{CIT}}_{x'|p}$ called intensities. A path ρ from the root to a leaf induces a part p, which is the intersection of the parts on the edges of the path: $p = \bigcap_{E(V_i,V_j)\in\rho} p_{V_j}$. The parts corresponding to paths of a CIT form a partition over $\mathcal{P}$, which can be shown easily using induction and the fact that the maps π_{V_i} induce disjoint parts p_{V_j} that cover $\mathcal{P}$.

A *Conditional Intensity Forest* (CIF) $\mathcal{F}_{x'}$ is a set of CITs $\{f_{x'}\}$. Because the parts of each CIT form a partition, a CIF induces a joint partition over $\mathcal{P}$ where a part p is the set of states s that have the same paths through all CITs. Finally, a CIF produces intensities from joint states by taking the product over the intensity factors from each CIT: $q^{\text{CIF}}_{x'|p^{\text{CIF}}} = \prod_{f_{x'}} q^{\text{CIT}}_{x'|p^{\text{CIT}}}$.

Using regression trees and forests can greatly reduce the number of model parameters. In CTBNs, the number of parameters grows exponentially in the number of parents per node. In tree and forest CTBNs, the number of parameters may be linear in the number of parents per node, exploiting the efficiency of using partitions. Notably, however, tree CTBNs are limited to having one intensity per parameter. In forest CTBNs, the number of intensities can be exponential in the number of parameters. Thus, the forest model has much greater potential expressivity per parameter than the other models.

Forest CTBN learning

Here we discuss the reasoning for using the multiplicative assumption and derive the changes in likelihood given modifications to the forest structure. Previous forests learners have used an additive assumption, e.g. averaging and aggregating, thereby taking advantage of properties of ensembles (Freund & Schapire, 1995; Breiman, 2001). However, if we take the sum over the intensity factors from each tree, there are no direct methods for calculating the change in likelihood aside from calculating the likelihood before and after a forest modification, which would require scanning the full data once per modification proposal. Furthermore, summing intensity factors could lead to intensities outside the valid domain $[0, \infty)$.

Instead we use a multiplicative assumption since it gives us the correct range over intensities. As we show below, using the multiplicative assumption also has the advantage that it is easy to compute the change in log likelihood with changes in forest structure. Consider a partition-based CTBN $\mathcal{M} = (\mathcal{B}, \{\mathcal{F}_{x'}\})$ where the partitions $P_{x'}$ and intensities $q_{x'|p}$ are given by the CIFs $\{\mathcal{F}_{x'}\}$. We focus on change in forest structure for one state $x' \in X$ and remove x' from the subscript notation for simplicity. Given a current forest structure $\mathcal{F}$ and its partition P, we formulate the change in likelihood by adding a new CIT f' and its partition P'. One example of f' is a new a one-split stub. Another example of f' is a tree copied to have the same structure as a CIT f in $\mathcal{F}$ with all intensity factors set to one, except at one leaf node where a split is added. This is equivalent to adding a split to f. We denote $\hat{P}$ as the joint partition of P and P' and parts $\hat{p} \in \hat{P}$, $p \in P$, and

$p' \in P'$. We consider the change in log likelihood ΔLL given the new and old models:

$$\begin{aligned}
\Delta LL &= (\sum_{\hat{p}} M_{\hat{p}} \log q_{\hat{p}} - q_{\hat{p}} T_{\hat{p}}) - (\sum_{p} M_p \log q_p - q_p T_p) \\
&= (\sum_{\hat{p}} M_{\hat{p}} (\log q_{p'} + \log q_p) - q_{\hat{p}} T_{\hat{p}}) - (\sum_{p} M_p \log q_p - q_p T_p) \\
&= (\sum_{\hat{p}} M_{\hat{p}} \log q_{p'} - q_{\hat{p}} T_{\hat{p}}) + \sum_{p} q_p T_p \\
&= \sum_{p'} M_{p'} \log q_{p'} - \sum_{\hat{p}} q_{\hat{p}} T_{\hat{p}} + \sum_{p} q_p T_p
\end{aligned} \tag{4.4}$$

We make use of the multiplicative assumption that $q_{\hat{p}} = q_{p'} q_p$ and $\sum_p M_p = \sum_{p'} M_{p'} = \sum_{\hat{p}} M_{\hat{p}}$ to arrive at equation 4.4. The first and third terms are easy to compute given the old intensities and new intensity factors. The second term is slightly more complicated:

$$\sum_{\hat{p}} q_{\hat{p}} T_{\hat{p}} = \sum_{\hat{p}} q_{p'} q_p T_{\hat{p}} = \sum_{p'} q_{p'} \sum_{\hat{p} \sim p'} q_p T_{\hat{p}}$$

We introduce the notation $\hat{p} \sim p'$ to denote the parts $\hat{p}$ that correspond to the part p'. The second term is a summation over parts $\hat{p}$; we have simply grouped together terms by membership in p'.

The number of parts in the joint partition set $\hat{P}$ can be exponentially large, but the only remaining dependency on the joint partition space in the change in log likelihood is the term $\sum_{\hat{p} \sim p'} q_p T_{\hat{p}}$. We can keep track of this value as we progress through the trajectories, so the actual time cost is linear in the number of trajectory intervals. Thinking of intensities q as rates, and given durations T, we observe that the second and third terms in equation 4.4 are expected numbers of transitions: $E_{\hat{p}} = \sum_{\hat{p}} q_{\hat{p}} T_{\hat{p}}$ and $E_p = \sum_p q_p T_p$. We additionally define $E_{p'} = \sum_{\hat{p} \sim p'} q_p T_{\hat{p}}$. Specifically, the expectations $E_{p'}$ and E_p are the expected number of transitions in part p' and p using the old model intensities, respectively, whereas $E_{\hat{p}}$ is the expected number of transitions using the new intensities.

Maximum-likelihood parameters

The change in log likelihood is dependent on the intensity factor values $\{q_{p'}\}$ we choose for the new partition. We calculate the maximum likelihood parameters by setting the derivative with respect to these factors to zero to get $q_{p'} = \frac{M_{p'}}{\sum_{\hat{p} \sim p'} q_p T_{\hat{p}}} = \frac{M_{p'}}{E_{p'}}$. Following the derivation in (Nodelman et al., 2003), we assign priors to the sufficient statistics calculations. Note, however, that the priors affect the multiplicative intensity factors, so a tree may split on the same partition set twice to get a stronger effect on the intensity, with the possible risk of undesirable overfitting.

Forest implementation

We use greedy likelihood maximization steps to learn multiplicative forests (mfCTBNs). Each iteration requires repeating three steps: (re)initialization, sufficient statistics updates, and model updates. Initially we are given a blank forest $\mathcal{F}_{x'}$ per state x' containing a blank tree $f_{x'}$, that is, a single root node acting as a leaf with an intensity factor of one. We also are given sets of possible splits $\{\sigma\}$ and a penalty function $\kappa(|Z|, |\mathcal{M}|)$ to penalize increased model complexity. First, for every leaf l in $\mathcal{M}$, we (re)initialize the sufficient statistics M_l and E_l in $\mathcal{M}$, as well as sufficient statistics for potential forest modifications: $M_{l,\sigma}$, $E_{l,\sigma}$, $\forall l, \sigma$. Then, we

traverse each of our trajectories $z \in Z$ to update each leaf. For every (state, duration) pair (s_i, t_i), where t_i is the time spent in state s_{i-1} before the transition to s_i, we update the sufficient statistics that compose equation 4.4. Finally, we compute the change in likelihood for possible forest modifications, and choose the modification with the greatest score. If this score is greater than the cost of the additional model complexity, κ, we accept the modification. We replace the selected leaf with a branch node split upon the selected σ. The new leaf intensity factors are the product of the old intensity (factor) q_l and the intensity factor $q_{p'}$. We present pseudocode in Algorithm 1.

Algorithm 1 Multiplicative forest learning

Input: trajectories $z_i \in Z$, blank forests $\mathcal{F}_{x'} \in \mathcal{M}$, partition sets $\{\Pi\}$, penalty $\kappa = \kappa(|Z|, |\mathcal{M}|)$

function LEARNMODEL(Z,$\mathcal{M}$)
  repeat
    resetSufficientStatistics($\mathcal{M}$)
    updateSufficientStatistics($Z, \mathcal{M}, \Pi$)
    zeroSplits = $makeSplits(\mathcal{M})$
  until zeroSplits = $true$
end function
function UPDATESUFFICIENTSTATISTICS(Z,$\mathcal{M}$,Π)
  for $(s_i, t_i) \in z_i \in Z$ **do**
    for Leaf $l = l_{j,x'}, \{l \in \{f_{x'}\} \in \mathcal{M} \mid d(l, s_i) = \emptyset\}$ **do**
      if($d(s_i, s_{i-1}) = \{x'\}$): $M_l = M_l + 1$
      $E_l = E_l + q_{x'|s_{i-1}} t_i$
      for $\pi \in \{\Pi\}$ **do**
        if($d(\pi, s_i) = \emptyset$, $d(s_i, s_{i-1}) = \{x'\}$): $M_{l,\pi} = M_{l,\pi} + 1$
        if($d(\pi, s_i) = \emptyset$): $E_{l,\pi} = E_{l,\pi} + q_{x'|s_{i-1}} t_i$
      end for
    end for
  end for
end function
function MAKESPLITS($\mathcal{M}$)
  madeSplit = $false$
  for $\{f_{x'}\} \in \mathcal{M}$ **do**
    Splits $\{\sigma_{l,\Pi}\} = \{(M_l, E_l, \{M_{l,\pi}, E_{l,\pi}\} \forall \pi \in \Pi)\}, \forall \Pi, l \in \{f_{x'}\}$
    (bestScore, bestSplit) = ($argmax_{\sigma_{l,\Pi}} \{deltaLogLikelihood(\sigma_{l,\Pi}) - \kappa\}$, $\sigma_{l,\Pi}$)
    if ($bestScore > 0$) **then**
      $split(\sigma_{l,\Pi})$
      if($\neg \mathcal{F}_{x'}.lastTreeBlank()$): $\mathcal{F}_{x'}.addBlankTree()$
      madeSplit = $true$
    end if
  end for
  return $\neg madeSplit$
end function
function DELTALOGLIKELIHOOD($\sigma_{l,\Pi}$)
  $q_\pi = M_\pi / E_\pi, \forall \pi \in \Pi$
  return $(\sum_\Pi M_\pi \log q_\pi - q_\pi E_\pi) + E_l$
end function

Unlike most forest learning algorithms, mfCTBNs learn trees neither in series nor in parallel. Notably, the best split is determined solely by the change in log likelihood, regardless of the tree to which it belongs. If it belongs to the blank tree at the end of the forest, that tree produces non-trivial factors and a new blank tree is appended to the forest. In this way, as mfCTBN learns, it automatically determines the forest size and tree depth according to the evidence in the data.

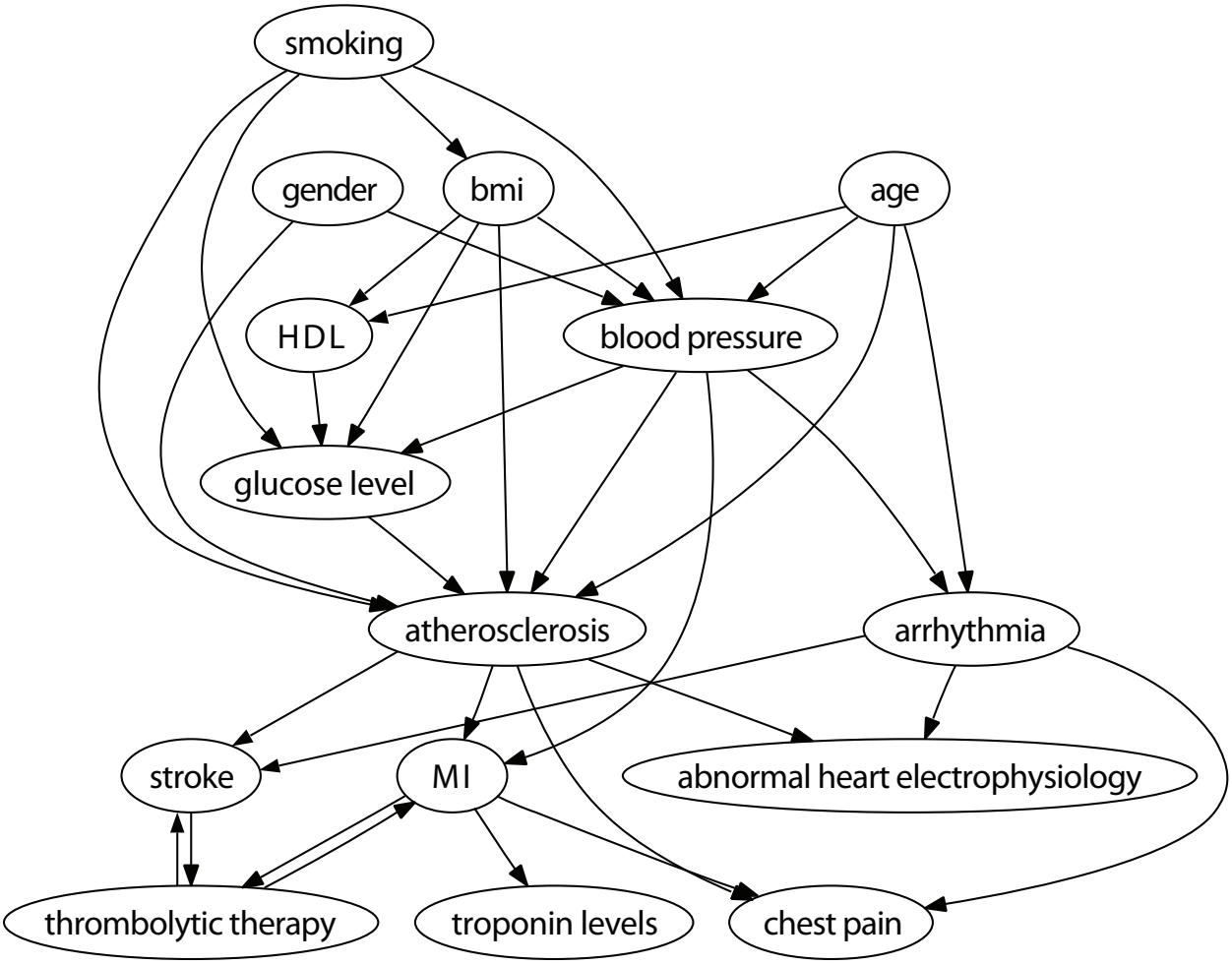

Figure 4.1: The cardiovascular health (CV health) structure used in experiments.

4.4 Experiments

We evaluate our tree learning and forest learning algorithms on samples from three models. The first model, which we call "Nodelman", is the benchmark model developed in (Nodelman, 2007; Nodelman et al., 2003). The second is a cardiovascular health model we call "CV health" shown in Figure 4.1. The cause of pathologies in this field are known to be multifactorial (Kannel, 1996). For example, it has been well-established that independent positive risk factors for atherosclerosis include being male, a smoker, in old age, having high glucose, high BMI, and high blood pressure. The primary tool for prediction in this field is risk factor analysis, where transformations over the *product* of risk factor values determines overall risk. The third model we call "S100" is a large-scale model with one hundred binary variables. Parents are determined by the binomial distribution $B(0.05, 200)$ over variable states, with intensity factor ratios of $1 : 0.5$. Our goal is to show that treeCTBNs and mfCTBNs can scale to much larger model types and still learn effectively. In our experiments we set the potential splits $\{\sigma\}$ to be the set of binary splits determined by indicators for each variable state x'. We set κ to be zero and terminate model learning when the tune set likelihood begins to decrease.

We compare our algorithms against the learning algorithm presented in (Nodelman et al., 2003) using

code from (Shelton et al., 2010), which we will call N-CTBN. N-CTBNs perform a greedy Bayesian structure search, adding, removing, or reversing arcs to maximize the Bayesian information criterion score, a trade-off between the likelihood and a combination of parameter and data size. Our algorithms use a tune set by sieving off one quarter of the original training set trajectories. We use the same Laplace prior as used in (Shelton et al., 2010). We use the same training and testing set for each algorithm. The trajectories are sampled from the ground truth models for durations $10, 10$ and 2 units of time, respectively. We evaluate the three models using the testing set average log likelihood. To provide an experimental comparison of model performance, we choose to analyze the p-values for a two-sided paired t-test for the average log likelihoods between mfCTBNs and N-CTBNs for each training set size. The results come from testing sets with one thousand sampled trajectories.

Results

Figure 4.2 (top) shows that the mfCTBN substantially outperforms both the treeCTBN and the N-CTBN on the Nodelman model in terms of average log likelihood. This effect is most pronounced with relatively few trajectories, suggesting that mfCTBNs are able to learn more quickly than either of the other models.

We observe an even larger difference between the mfCTBN and the other models in the CV health model in Figure 4.2 (middle). With relatively few trajectories, the mfCTBN is able to identify the multifactorial causes as observed in the high log likelihood and structural recall. For runs with fewer than 500 training set trajectories, many N-CTBN models have nodes including every other node as a parent, requiring the estimation of about 300,000 parameters on average.

Figure 4.2 (bottom) shows that mfCTBNs can effectively learn dense models an order of magnitude larger than those previously studied. The expected number of parents per node in the S100 model is approximately 20. In order to exactly reconstruct the S100 model, a traditional CTBN would then need to estimate 2^{21} intensity values. For many applications, variables need more parents than this. We observe that N-CTBNs have difficulty scaling to models of this size. The N-CTBN learning time on this data set ranges from 4 hours to more than 3 days; runs were stopped if they had not terminated in that time. About one third of the runs failed to complete, and the runs that did complete suggested that N-CTBN performed poorly, similar to the differences observed in the CV health experiment. We suspect the algorithm may be similarly building nodes with many parents; the model might need to estimate 2^{100} parameters, a bottleneck at minimum. By comparison, all runs using treeCTBNs and mfCTBNs completed in less than 1 hour. The averaged results of N-CTBNs on the S100 model are omitted accordingly.

We tested for significant differences in the average log likelihoods between the N-CTBN and mfCTBN learning algorithms. In the Nodelman model, the differences were significant at level of $p =$1e-10 for sizes 10 through 500, $p = 0.05$ for sizes 1000 and 5000, and not significant for size 10000. In the CV health model, the differences were significant at $p =$1e-9 for all training set sizes. We were unable to generate a t-test comparison of the S100 model.

Figure 4.3 shows the ground truth forest and the mfCTBN forest learned for the "severe atherosclerosis" state in the CV health model. To calculate the intensity of transitioning *into* this state, we identify the leaf in each forest that matches the current state and take the product of their intensity factors. Figure 4.3 (bottom) shows the recovery of the correct dependencies in approximately the right ratios.

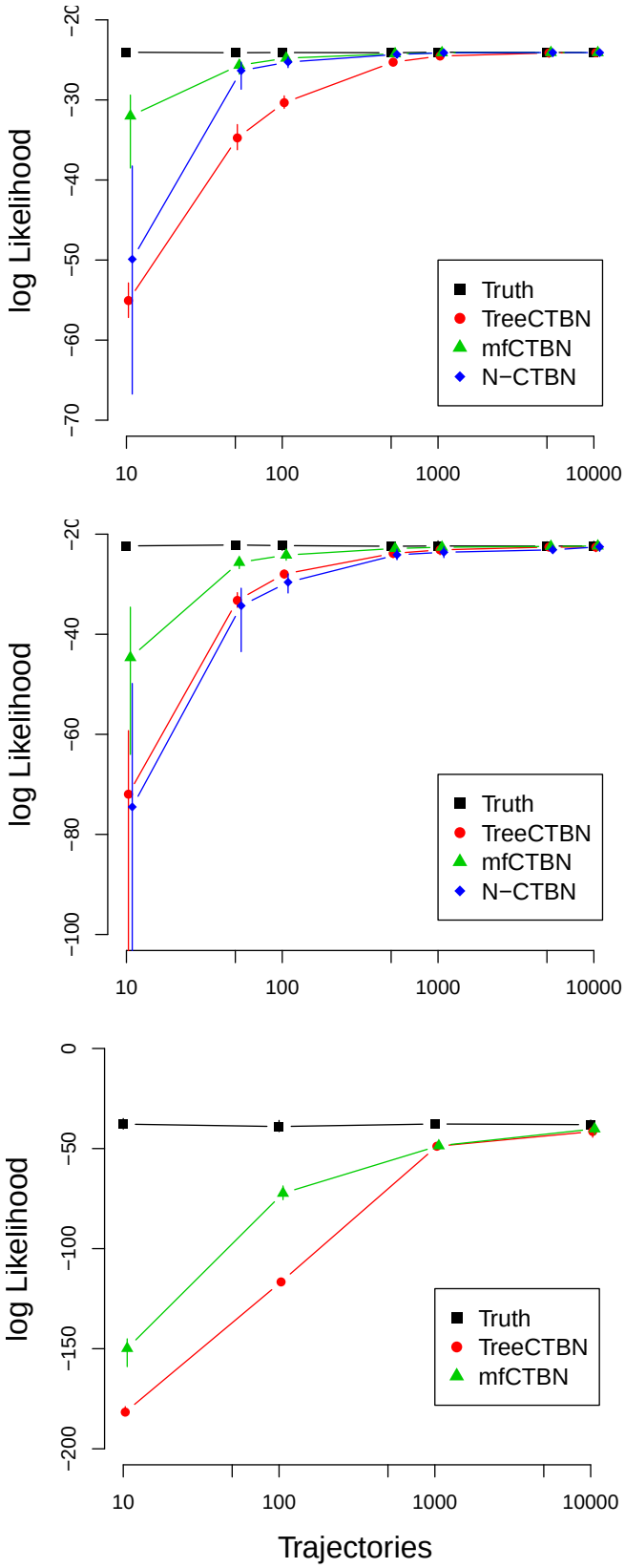

Figure 4.2: Average testing set log likelihood varying the training set size for each model: Nodelman (top), CV health (middle), and S100 (bottom). N-CTBN averages are omitted on the S100 model as one third of the runs did not terminate.

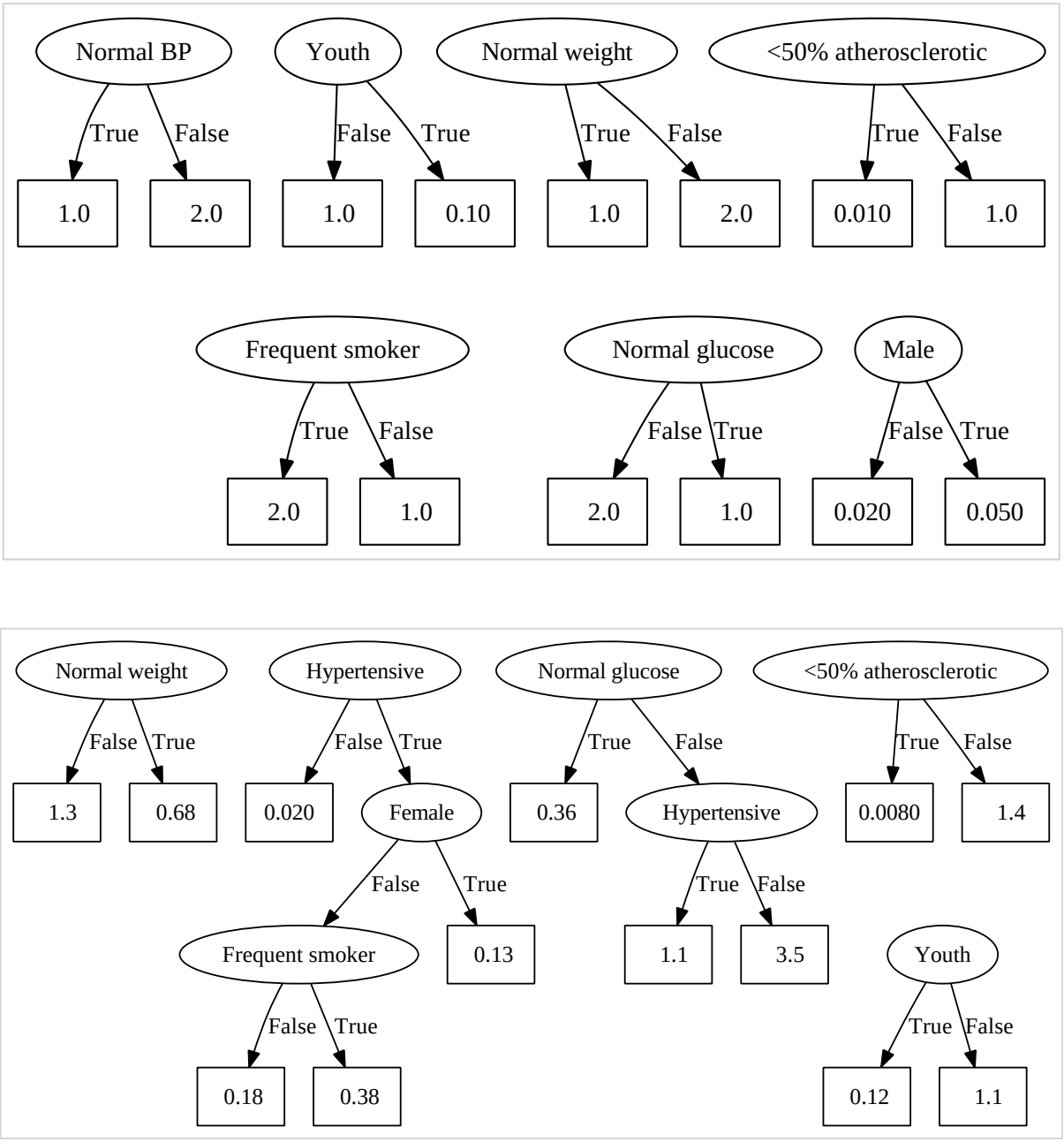

Figure 4.3: Ground truth (top) and mfCTBN forest learnt from 1000 trajectories (bottom) for intensity/rate of developing severe atherosclerosis.

Learning curves

To characterize basic properties of multiplicative forest learning, we investigated the importance of the tune set as the stopping criterion. Figure 4.4 shows the learning curves for training and testing sets as a function of split attempts. Vertical lines show where the BIC and AIC scores would have terminated the process: $\kappa(BIC) = \frac{1}{2}|H| \log |Z|$ and $\kappa(AIC) = |H|$. Unlike forests used in ensembles, multiplicative forests do not exhibit stabilizing behavior. We suspect that the decreased model stability as the number of split attempts

increases might be due to the multiplicative assumption.

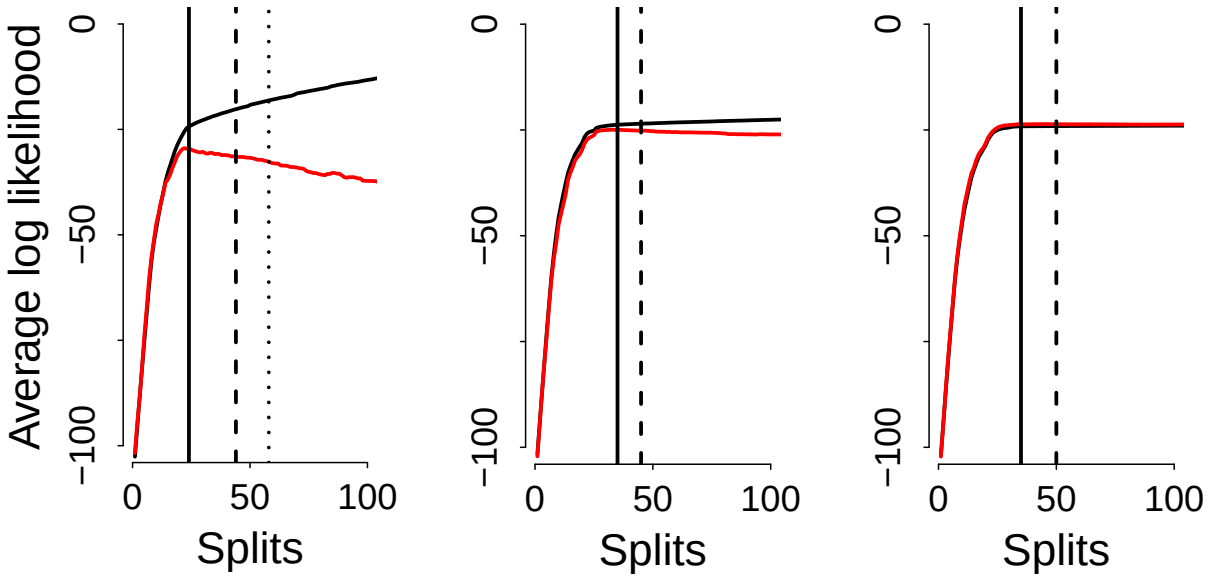

Figure 4.4: Training and testing set average log likelihood (black and red, respectively), for training set size of 10, 100, 1000 on the CV health model. {Solid, dashed, dotted} vertical lines indicate the {tune set, BIC, and AIC} stopping criterion if met.

4.5 Related Work

We discuss the relationships between mfCTBNs and related work in two areas: forest learning and continuous-time processes. Forest learning with a multiplicative assumption is equivalent to forest learning in the log space with an additive assumption and exponentiating the result. This suggests that our method shares similarities with functional gradient boosting (FGB), a leading method for constructing regression forests, run in the log space (Friedman, 2001).

Specifically, in Section 4.3, we showed that, given a new partition proposal p', the maximum likelihood intensity factors are given by the ratio of the observed to expected number of transitions: $M_{p'}/E_{p'}$.

$$\frac{\text{Observed transitions in } Z}{\text{Expected transitions under } H} = \frac{M_{p'}}{E_{p'}}$$

This result suggests a connection to functional gradient boosting (FGB), one of the leading methods for constructing regression forests (Friedman, 2001). FGB methods perform gradient-descent in the function space by fitting regression trees to residuals at every gradient step. Suppose we have observations y on the domain $[0, \infty)$; we might use FGB to learn $\log y$ because FGB uses additive trees, and directly learning y from x could give negative values, i.e. $\hat{y} = f(x)$ outside the domain. Using FGB over $\log y|x$ builds multiplicative forests: the residual predicted in the tree $f_{i+1}(x)$ is $(\log y - \sum_{f_i} f_i(x))$, and taking the exponent of this quantity is simply the ratio $y/\hat{y}$.

Nevertheless, there are several critical differences between mfCTBNs and FGB learning. First, mfCTBNs are not given explicit outcomes y, so updates maximize the change in log likelihood based on sufficient

statistics calculations instead of minimizing a loss function in FGB. Second, our algorithm does not restrict learning of additional trees prior to the completion of previous trees, allowing the model to determine when to expand the forest size or tree depth. Node splits in any tree can occur in any iteration of forest learning. By comparison, in FGB, trees are constructed to completion and are static as new trees are learned. To provide the ability to modify any tree at any learning iteration, FGB would have to do leave-one-(tree)-out modeling, that is, predict $\log y - \sum_{f_i, i \neq j} f_i(x)$ for all j, a potentially expensive operation. To recap, our method is different primarily in its direct use of a likelihood-based objective function and in its ability to modify any tree in the forest at any iteration.

Several other works that model variable dependencies over continuous time also exist. Poisson process networks and cascades model variable dependencies and event rates (Rajaram et al., 2005; Simma, 2010). Perhaps the most closely related work, piecewise-constant conditional intensity models (PCIMs), reframes the concept of a factored CTMP to allow learning over arbitrary basis state functions with trees, possibly piecewise over time (Gunawardana et al., 2011). These point process models focus on the "positive class", i.e. the observation or count of observations of an event. Thus they run into the limitations of making the closed-world assumption. That is, given a timeline, we receive all *observations* of events but not necessarily all *occurrences* of the events, and we would like to include this uncertainty in our model. In point processes, the representation of the "negative" class is missing, when in some cases it is the absent state of a variable that triggers a process, as for example in the case of gene expression networks and negative regulation. Nonetheless, in Chapter 6 we extend the multiplicative forest idea to sidestep the inference problems that are discussed in the next chapter.

4.6 Summary

We presented an alternative representation of the dynamics of CTBNs using partition-based CTBNs instantiated by trees and forests. Our models grow linearly in the number of forest node splits, while CTBNs grow exponentially in the number of parent nodes per variable. Motivated by the domain over intensities, we introduced multiplicative forests and showed that CTBN likelihood updates can be efficiently computed using changes in log likelihood. Finally, we showed that mfCTBNs outperform both treeCTBNs and N-CTBNs in three experiments and that mfCTBNs are scalable to problems with many variables. With our contributions in developing scalable CTBNs and efficient learning, along with continued improvements in inference, CTBNs can be a powerful statistical tool to model complex processes over continuous time. We expose the challenges of CTBN inference in the next chapter and develop a sampling method that improves upon the existing sequential importance sampler, in turn improving the scalability of CTBN inference to problems where more evidence is observed.

5 Rejection-Based Inference for Continuous-Time Bayesian Networks

Overview

Having presented an efficient representation and learning algorithm for CTBNs in the previous chapter, in this chapter we discuss how to approach timelines with incomplete observations. Approximate inference procedures based on sequential importance sampling are often used, but when proposal and target distributions are dissimilar, the procedures lead to biased estimates or require a prohibitive number of samples. This chapter introduces a method that better approximates the target distribution by sampling variable by variable from existing importance samplers and accepting or rejecting each proposed assignment in the sequence: a choice made based on anticipating upcoming evidence. We relate the per-variable proposal and target distributions by expected weight ratios of sequence completions and show that we can learn accurate models of optimal acceptance probabilities from local samples. In a continuous-time domain, our method improves upon previous importance samplers by transforming a sequential importance sampling problem into a machine learning one. A similar version of this chapter is in preparation for submission.

5.1 Introduction

Sequential importance sampling (SIS) is a method for approximating a target distribution that samples from a proposal distribution and weights by the ratio of target and proposal distributions at each step of the sequence. It provides the basis for many distribution approximations with applications including robotic environment mapping and speech recognition (Montemerlo et al., 2003; Wolfel & Faubel, 2007). The characteristic shortcoming of importance sampling stems from the potentially high weight variance that results from large differences in the target and proposal densities. SIS compounds this problem by iteratively sampling over the steps of the sequence, resulting in sequence weights that are the product of step weights. The sequence weight distribution is exponential, so only the high-weight tail of samples contributes substantially to the distribution approximation. Two approaches to mitigate this problem are filtering, *e.g.*, (Doucet et al., 2000; Fan et al., 2010), where particles are resampled according to their weights to maintain a low-variance weight distribution, and adaptive importance sampling, *e.g.*, (Cornebise et al., 2008; Yuan & Druzdzel, 2003; 2007a), where the proposal distribution adapts to be closer to the target.

One drawback of filtering is that it does not efficiently account for future evidence, and in cases of severe proposal-evidence mismatch, many resampling steps are required, leading to sample impoverishment. In line with adaptive importance sampling, our method addresses proposal-evidence mismatch by developing "foresight", *i.e.* adaptation to approaching evidence, to guide its proposals. It develops "foresight" by learning a binary classifier dependent on approaching evidence to accept or reject the step from the original proposal distribution. Our procedure can be viewed as the construction of a second proposal distribution, learned to account for evidence and to better approximate the target distribution. Our method is a new form of adaptive importance sampling; we contrast our method with earlier forms in Section 5.1.

In greater detail, our task is to recover a target distribution f^*, which can be factored variable by variable into component conditional distributions f_i^* for $i \in 1 \dots k$. The SIS framework provides a suboptimal

surrogate distribution g, which likewise can be factored into a set of conditional distributions g_i. We propose a second surrogate distribution h closer to f^* based on learning conditional acceptance probabilities a_i of rejection samplers relating f_i^* and g_i. That is, to sample from h, we iteratively (re-)sample from proposals g_i and accept with probability a_i.

Our key idea is to relate the proposal g_i and target f_i^* distributions by the ratio of expected weights of sequence completions, *i.e.*, a setting for each variable from i to k, given acceptance and rejection of the sample from g_i. Given the expected weight ratio, we can recover the optimal acceptance probability a_i^* and thus f_i^*.

Unfortunately, the calculation of the expected weights, and thus the ratio, is typically intractable because of the exponential number of sequence completions. Instead, we can approximate it using machine learning. First, we show that the expected weight ratio equals the odds of accepting a proposal from g_i under the f_i^* distribution. Then, transforming the odds to a probability, we can learn a binary classifier for the probability of acceptance under f_i^* given the sample proposal from g_i. Finally, we show how to generate examples to train a classifier to make the optimal accept/reject decision.

We specifically examine the application of our rejection-based SIS algorithm to continuous-time Bayesian networks (CTBNs) (Nodelman, 2007), which have applications for example in anomaly detection (Xu & Shelton, 2010) and medicine (Weiss et al., 2012c). We find our methods to be more generally applicable, *e.g.*, to dynamic Bayesian networks and sequential forecasting models, but we focus our analysis on CTBNs. The existing CTBN importance sampler g uses a combination of exponential and truncated exponential distributions to select interval transitions that agree with evidence (Fan et al., 2010). Using g, each evidence point causes a stochastic downweighting in a fraction of the samples, which results in an increase in variance of the importance weights and exhibits a mismatch between f^* and g. Because a sequence weight corresponds to the product of its interval weights, the stochastic downweighting of intervals approaching non-matching evidence produces a high-variance distribution of sequence weights. Experimentally, we show that rejection-based SIS improves our ability to approximate f^* with many fewer samples.

We proceed as follows. We conclude the introduction with an illustrative example and related work. In Section 5.2, we define rejection sampling within sequential importance sampling and show how to approximate the target distribution via binary classification. In Section 5.3, we extend our analysis to continuous-time Bayesian networks. We describe experiments in Section 5.4 that show the empirical advantages of our method over previous CTBN importance samplers. Possible extensions related to our work are provided in Section 5.5 and we conclude in Section 5.6.

An Illustrative Example

Figure 5.1 describes our method in the simplest relevant example: a binary-state Markov chain. For our example, let $k = 3$: then we have evidence that $z_3 = 1$. One possible sample procedure could be:

$$S,\ \text{accept } z_1^2,\ \text{reject } z_2^2,\ \text{accept } z_2^1,\ \text{reject } z_3^2,\ \text{accept } z_3^1, T,$$

giving us the path: $S,\ z_1^2,\ z_2^1,\ z_3^1,\ T$. Note that if the proposal g_3 to z_3^1 given state z_2^1 were very improbable under g but not f (*i.e.*, proposal-evidence mismatch), all samples running through z_2^1 would have very large weight. By introducing the possibility of rejection at each step, our procedure can learn to reject samples to z_3^2, reducing the importance sampling weight, and learn to enter states z_2^1 and z_2^2 proportionally to $f(\cdot|e)$, *i.e.*, develop "foresight".

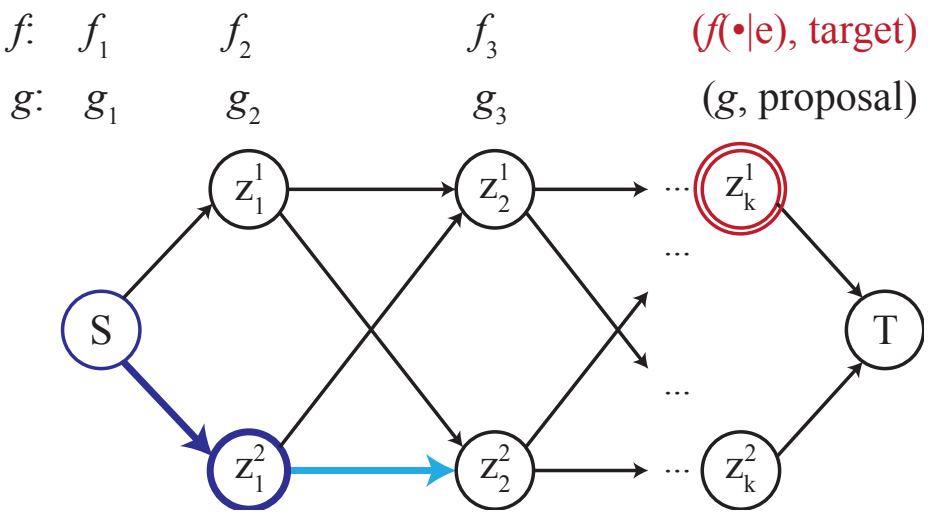

Figure 5.1: A source-to-sink representation of a binary-state Markov chain with evidence at z_k (red). Distributions f and g are defined over paths from source S to sink T and are composed of element-wise distributions f_i and g_i. If a sample is at state z_1^2 (dark blue), an assignment to z_2^2 is proposed (light blue) according to g_2. To mimic sampling from $f_2^* = f_2(\cdot|e)$, the proposed assignment is accepted with probability proportional to the ratio of expected weights of path completions from z_2^2 and z_1^2 to T, giving us our proposal h_2.

Related Work

As mentioned above, batch resampling techniques based on rejection control (Liu et al., 1998; Yuan & Druzdzel, 2007b) or sequential Monte Carlo (SMC) (Doucet et al., 2000; Fan et al., 2010), *i.e.* particle filtering, can mitigate the SIS weight variance problem, but they can lead to reduced particle diversity, especially when many resampling iterations are required. Particle smoothing (Fan et al., 2010) combats particle impoverishment, but the exponentially-large state spaces used in CTBNs limit its ability to find alternative, probable sample histories. Previous adaptive importance sampling methods rely on structural knowledge and other inference methods, *e.g.*, (Cheng & Druzdzel, 2000; Yuan & Druzdzel, 2003), to develop improved proposals, whereas our method learns a classifier to help guide samples through regions of proposal-evidence mismatch. One interesting idea combining work in filtering and adaptive importance sampling is the SMC^2 algorithm (Chopin et al., 2011), which maintains a sample distribution over both particles and parameters determining the proposal distribution, resampling along either dimension as necessary. The method does not anticipate future evidence, so it may complement our work, which can similarly be used in the SMC framework. Other MCMC (Rao & Teh, 2011) or particle MCMC (Andrieu et al., 2010) methods may have trouble in large state spaces (*e.g.*, CTBNs) with multiple modes and low density regions in between, especially if there is proposal-evidence mismatch.

5.2 Learning to Reject

Recall from the importance sampling setup that we have two distributions f and g, and the existing sampling approach approximates f with samples from g. Each sample from g comes with an associated weight w_{fg}.

Now, we design a second surrogate $h(z)$ with the density corresponding to accepting a sample from g:

$$h(z) = g(z)a(z)\left(\int_{\Omega} g(\zeta)a(\zeta)d\zeta\right)^{-1} \tag{5.1}$$

where $a(z)$ is the acceptance probability of the sample from g. The last term in Equation 5.1 is a normalizing functional (of g and a) to ensure that $h(z)$ is a density. Procedurally, we sample from h by (re-)sampling from g and accepting with probability a. The approximation of f^* with h is given by:

$$\begin{aligned}\int_{\mathcal{Z}} f^*(z)dz &= \int_{\mathcal{Z}} \frac{f^*(z)}{g(z)}\frac{g(z)}{h(z)}h(z)dz \\ &\approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[z^i \in \mathcal{Z}] w^i_{f^*g} w^i_{gh}\end{aligned}$$

with weights $w^i_{f^*h} = w^i_{f^*g} w^i_{gh}$. To ensure that h has the support of f^*, we require that both a and g are non-zero everywhere f^* is non-zero.

Now we can define our optimal resampling density $h^*(z)$ using the optimal choice of acceptance probability $a^*(z) = \min(1, f^*(z)/\alpha g(z))$, where $\alpha \geq 1$ is a constant determining the familiar rejection sampler "envelope": $\alpha g(z)$. The density $h^*(z)$ is optimal in the sense that, for appropriate choice of α such that $f^*(z) < \alpha g(z)$ for all z, $h^*(z) = f^*(z)$. When $h^*(z) = f^*(z)$, the importance weights are exactly 1, and the effective sample size is n.

In many applications the direct calculation of $f^*(z)$ is intractable or impossible and thus we cannot directly recover $a^*(z)$ or $h^*(z)$. However, we can still use these ideas to find $h^*(z)$ through sequential importance sampling (Liu et al., 1998), which we describe next.

Now, we consider the sequential importance sampling (SIS) extension, where we again need to identify the relationship between the target and proposal decompositions. Recall that we are interested in sampling directly from the conditional distribution $f^*(z) = f(z|e)$ for fixed e. We define interval distributions $f^*_1(z_1)$ and $f^*_i(z_i|z_{(i-1)\leftarrow 1})$ such that $f^*(z)$ can be factored into the interval distributions: $f^*(z) = f^*_1(z_1)\prod_{i=2}^{k} f^*_i(z_i|z_{(i-1)\leftarrow 1})$. Using Bayes' theorem, we have:

$$f^*(z) = \frac{p(e|z_1)}{p(e)}p(z_1)\prod_{i=2}^{k}\frac{p(e|z_{i\leftarrow 1})}{p(e|z_{(i-1)\leftarrow 1})}p(z_i|z_{(i-1)\leftarrow 1}).$$

Thus, we define the interval distributions:

$$f^*_i(z_i|z_{(i-1)\leftarrow 1}) = \frac{p(e|z_{i\leftarrow 1})}{p(e|z_{(i-1)\leftarrow 1})}p(z_i|z_{(i-1)\leftarrow 1})$$

for $i > 1$ and $p(e|z_{(i-1)\leftarrow 1}) > 0$, and $f^*(z_1) = p(e|z_1)p(z_1)/p(e)$ for $i = 1$ and $p(e) > 0$. Then, by the law of probability, we have:

$$f^*_i(z_i|z_{(i-1)\leftarrow 1}) = \frac{\mathbb{E}_f[\mathbb{1}[e,z]|z_{i\leftarrow 1}]}{\mathbb{E}_f[\mathbb{1}[e,z]|z_{(i-1)\leftarrow 1}]}p(z_i|z_{(i-1)\leftarrow 1}). \tag{5.2}$$

The indicator function $\mathbb{1}[e,z]$ is shorthand for $\mathbb{1}[\bigcap_{l\in\kappa}\{z_l{=}e_l\}|z]$ and takes value 1 if the evidence matches z and 0 otherwise. Note that in the sampling framework, Equation 5.2 corresponds to sampling from the unconditioned proposal distribution $p(z_i|z_{(i-1)\leftarrow 1})$ and calculating the expected weight of sample completions $z_{k\leftarrow(i+1)}$ and $z_{k\leftarrow i}$, given by the indicator functions. This procedure describes the expected outcome obtained by forward sampling with rejection.

However, when f^* and f are highly dissimilar, the vast majority of samples from f will be rejected, *i.e.*, $\mathbb{1}[e, z] = 0$ for most z. It may be better to sample from a proposal g with weight function $w = f/g$ so that sampling leads to fewer rejections. Substituting g in Equation 5.2, we get:

$$f_i^*(z_i|z_{(i-1)\leftarrow 1}) = \frac{\mathbb{E}_g[w^a_{k\leftarrow i}]}{\mathbb{E}_g[w^r_{k\leftarrow i}]} g_i(z_i|z_{(i-1)\leftarrow 1}). \tag{5.3}$$

The terms $\mathbb{E}_g[w^a_{k\leftarrow i}]$ and $\mathbb{E}_g[w^r_{k\leftarrow i}]$ are the expected forward importance sampling weights of $z_{k\to i}$ given acceptance (a) or rejection (r) of proposed assignment z_i.

To derive Equation 5.3, we relate the target densities $f_i^*(z_i|z_{(i-1)\leftarrow 1})$ with the proposal densities $g_i(z_i|z_{(i-1)\leftarrow 1})$ via the (standard) derivation of Equation 4 by employing Bayes' theorem, the law of total probability, and substitution:

$$\begin{aligned}
&f_i^*(z_i|z_{(i-1)\leftarrow 1}) \\
&= \frac{p(e|z_{i\leftarrow 1})}{p(e|z_{(i-1)\leftarrow 1})} p(z_i|z_{(i-1)\leftarrow 1}) \\
&= \frac{\sum_{z_{k\leftarrow i+1}} p(e|z_{k\leftarrow i+1}, z_{i\leftarrow 1}) p(z_{k\leftarrow i+1}|z_{i\leftarrow 1})}{\sum_{z_{k\leftarrow i}} p(e|z_{k\leftarrow i}, z_{i-1\leftarrow 1}) p(z_{k\leftarrow i}|z_{i-1\leftarrow 1})} p(z_i|z_{i-1\leftarrow 1}) \\
&= \frac{\sum_{z_{k\leftarrow i+1}} \mathbb{1}[\bigcap_{l\in\kappa}\{z_l{=}e_l\}|\{z\}] p(z_{k\leftarrow i+1}|z_{i\leftarrow 1})}{\sum_{z_{k\leftarrow i}} \mathbb{1}[\bigcap_{l\in\kappa}\{z_l{=}e_l\}|\{z\}] p(z_{k\leftarrow i}|z_{i-1\leftarrow 1})} p(z_i|z_{i-1\leftarrow 1}) \\
&= \frac{\mathbb{E}_g[\mathbb{1}[e, z] \prod_{j=i+1}^{k} w_j(z_j|z_{j-1\leftarrow 1})|z_{i\leftarrow 1}]}{\mathbb{E}_g[\mathbb{1}[e, z] \prod_{j=i}^{k} w_j(z_j|z_{j-1\leftarrow 1})|z_{i-1\leftarrow 1}]} p(z_i|z_{i-1\leftarrow 1}) \\
&= \frac{\mathbb{E}_g[\mathbb{1}[e, z] \prod_{j=i}^{k} w_j(z_j|z_{j-1\leftarrow 1})|z_{i\leftarrow 1}]}{\mathbb{E}_g[\mathbb{1}[e, z] \prod_{j=i}^{k} w_j(z_j|z_{j-1\leftarrow 1})|z_{i-1\leftarrow 1}]} g_i(z_i|z_{i-1\leftarrow 1}) \\
&= \frac{\mathbb{E}_g[w^a_{k\leftarrow i}]}{\mathbb{E}_g[w^r_{k\leftarrow i}]} g_i(z_i|z_{(i-1)\leftarrow 1}). \qquad \square
\end{aligned}$$

Equation 5.3 provides the relationship we want: f_i^* versus g_i, given by the ratio of expected weights of completion of sample z under acceptance or rejection of z_i. This allows us to further improve g and gives us the sampling distribution h.

Rejection Sampling to Recover f_i^*

Because Equation 5.3 relates the two distributions, we can generate samples from f_i^* by conducting rejection sampling from g_i. Selecting constant α such that $f_i^* \leq \alpha g_i$, *i.e.*, αg_i is the rejection envelope, we define the optimal interval acceptance probability a_i^* by:

$$a_i^*(z_i|z_{(i-1)\leftarrow 1}) = \frac{f_i^*(z_i|z_{(i-1)\leftarrow 1})}{\alpha g_i(z_i|z_{(i-1)\leftarrow 1})} = \frac{\mathbb{E}_g[w^a_{k\leftarrow i}]}{\alpha \mathbb{E}_g[w^r_{k\leftarrow i}]}. \tag{5.4}$$

By defining a_i^* for all i, we can generate an unweighted sample from $f^*(z)$ in $O(k \max_i(f_i^*(\cdot)/g_i(\cdot)))$ steps given the weight ratio expectations and appropriate choice of α. Thus, if we can recover a_i^* for all i, we get $h = f^*$ as desired and our procedure generates unweighted samples.

Estimating the Weight Ratio

The procedure of sampling intervals to completion depends on the expected weight ratio in Equation 5.3. Unfortunately, exact calculation of the ratio is impractical because the expectations involved require summing over an exponential number of terms. We could resort to estimating it from weighted importance samples: completions of z given $z_{i\leftarrow 1}$ and z given $z_{(i-1)\leftarrow 1}$. While possible, this is inefficient because (1) it would require weight estimations for every z_i given $z_{i\leftarrow 1}$, and (2) the estimation of the expected weights itself relies on importance sampling.

However, we can cast the estimation of the weight ratio as a machine learning problem of binary classification. We recognize that similar situations, in terms of state $z_{i\leftarrow 1}$, evidence e, model (providing f) and proposal g, result in similar values of a_i^*. Thus, we can learn a binary classifier $\Phi_i(z_{i\leftarrow 1}, e, f, g)$ to represent the probability of {acceptance, rejection} = $\{\phi_i(\cdot), 1-\phi_i(\cdot)\}$ as a function of the situation.

In particular, the expected weight ratio in Equation 5.3 is proportional to the odds under f^* of accepting the z_i sampled from g_i. The binary classifier provides an estimate of the probability of acceptance $\phi_i(z_{i\leftarrow 1}, e, f, g)$, from which we can derive the odds of acceptance. Substituting into Equation 5.4, we have:

$$a_i^*(z_i|z_{(i-1)\leftarrow 1}) \approx \frac{1}{\alpha}\left(\frac{\phi_i(z_{i\leftarrow 1}, e, f, g)}{1-\phi_i(z_{i\leftarrow 1}, e, f, g)}\right) = a_i(z_i|z_{(i-1)\leftarrow 1})$$

denoting the approximations as a_i for all i. Then our empirical proposal density h is:

$$\begin{aligned} h(z) &= h_1(z_1)\prod_{i=2}^{k} h_i(z_i) \\ &= g_1(z_1)a_1(z_1)c_1[g_1, a_1]\prod_{i=2}^{k} g_i(z_i|z_{(i-1)\leftarrow 1})a_i(z_i|z_{(i-1)\leftarrow 1})c_i[g_i, a_i] \end{aligned}$$

where the c_i are the normalizing functionals as in Equation 5.1. We provide pseudocode for the rejection-based SIS procedure in Algorithm 2.

Training the Classifier Φ_i

Conceptually, generating examples for the classifier Φ_i is straightforward. Given some $z_{(i-1)\leftarrow 1}$, we sample z_i from g_i, accept with probability $\rho = 1/2$, and sample to completion using g to get the importance weight. Then, an example is $(y, \mathbf{x}, w)$: $y =$ {accept, reject}, $\mathbf{x}$ is a set of features encoding the "situation", and w is the importance weight.

The training procedure works because the mean weight of the positive examples estimates $\mathbb{E}_g[w^a_{k\leftarrow i}]$, and likewise the mean weight of the negative examples estimates $\mathbb{E}_g[w^r_{k\leftarrow i}]$. By sampling with training acceptance probability ρ, a calibrated classifier Φ_i^ρ (*i.e.*, one that minimizes the L_2 loss) estimates the probability: $\rho\mathbb{E}_g[w^a_{k\leftarrow i}]/(\rho\mathbb{E}_g[w^a_{k\leftarrow i}] + (1-\rho)\mathbb{E}_g[w^r_{k\leftarrow i}])$. The estimated probability can then be used to recover the expected weight ratio:

$$\frac{\mathbb{E}_g[w^a_{k\leftarrow i}]}{\mathbb{E}_g[w^r_{k\leftarrow i}]} \approx \left(\frac{1-\rho}{\rho}\right)\left(\frac{\phi_i^\rho(z_{i\leftarrow 1}, e, f, g)}{1-\phi_i^\rho(z_{i\leftarrow 1}, e, f, g)}\right),$$

and thus, the optimal classifier can be used to recover the rejection-based acceptance probability a_i^*. We get our particular estimator $\phi_i/(1-\phi_i)$ by setting $\rho = 1/2$, though in principle we could use other values of ρ.

In practice, we sample trajectories alternating acceptance and rejection of samples z_i to get a proportion $\rho = 1/2$. Then, we continue sampling the same trajectory to produce $2k$ training examples for a sequence

Algorithm 2 Rejection-based SIS

Input: conditional distributions $\{f_j\}$ and $\{g_j\}$ $\forall j$, evidence e; constants α, k; $i = 1$, $z = \{\}$, $w = 1$; classifiers $\{\phi_j\}$
Output: sample z with weight w

function SAMPLEH(...)
  while $i \leq k$ **do**
    accept = `false`
    while `not` accept **do**
      Sample $z_i \sim g_i$, $r \sim \mathrm{U}[0, 1]$
      $a = \frac{1}{\alpha}\left(\frac{\phi_i(z_i,z,e,f,g)}{1-\phi_i(z_i,z,e,f,g)}\right)$
      if $r < a$ **then**
        accept = `true`
      end if
    end while
    $z = \{z_i, z\}, w = w f_i(z_i) c[g_i(z_i), a(z_i)]/(g_i(z_i)a)$
    $i = i + 1$
  end while
  return (z, w)
end function

of length k. We adopt this procedure for efficiency at the cost of generating related training examples. We provide pseudocode for the example generation procedure for learning the classifier in Algorithm 3.

Inevitably, there is some cost to pay to construct h, including time for feature construction, classifier training, and classifier use. The level of sophistication needed will be problem-dependent. However, in many challenging problems the simpler methods will not produce an ESS of any appreciable size, in our case because of a mismatch between f^* and g, and in MCMC because of mode hopping difficulties. Our method adopts an approach complementary to particle methods to help tackle such problems, and we show its utility in the CTBN application.

5.3 Sampling in CTBNs

Evidence provided in data is typically incomplete, *i.e.*, the joint state is partially or fully unobserved over time. Thus, inference is performed to probabilistically complete the unobserved regions. CTBNs are generative models and provide a sampling framework to complete such regions. Let a trajectory z be a sequence of (state,time) pairs $(z_i{=}\{x_{1i}, x_{2i}, \ldots, x_{di}\}, t_i)$ for $i = \{0, \ldots, k\}$, where x_{ji} is the jth CTBN variable at the ith time, such that the sequence of t_i are in $[t_{\text{start}}, t_{\text{end}})$. Given an initial state $z_0 = \{x_{10}, x_{20}, \ldots, x_{d0}\}$, transition times are sampled for each variable x according to $q_{x|u}e^{-q_{x|u}t}$ where x is the active state of X. The variable X_i that transitions in the interval is selected based on the shortest sampled transition time. The state to which X_i transitions is sampled from $\Theta_{x_i x_i'|u}$. Then the transition times are resampled according to intensities $q_{x|u}$, noting that these intensities may be different because of potential changes in the parents setting u. Due to the memoryless property of exponential distributions, no resampling of the transition time for a variable X is needed if the intensity $q_{x|u}$ is unchanged. The trajectory terminates when all sampled transition times exceed a specified ending time.

Previous work by Fan et al. describes a framework for importance sampling, particle filtering, and particle smoothing in CTBNs (Fan et al., 2010). The idea is to modify the sampling of each interval so that the interval end time cannot pass by impending, non-matching evidence. The process is as follows. The first future non-matching evidence states are found for each variable and their corresponding evidence

Algorithm 3 Training examples for learning

Input: conditional distributions $\{f_j\}$ and $\{g_j\}$ $\forall j$, evidence e^l for l in evidence data; n examples
Output: classifier ϕ

function GET_EXAMPLES($e, \{f_j\}, \{g_j\}, k$)
  $z = \{\}; i = 1; w = 1; d = 1$
  while $i \leq k$ **do**
    // one negative, one positive example //
    for $l \in$ {reject, accept} **do**
      Sample $z_i \sim g_i(z)$
      $w_i = f_i(z_i)/g_i(z_i)$
      $(y, x, u)_d = (l, \text{get\_x}(z_i, z, e, \{f_j\}, \{g_j\}), w_i)$
      $d = d + 1$
    end for
    $z = \{z_i, z\}; w = ww_i; i = i + 1$
  end while
  for $i = 1$ to d **do**
    $(y, x, u)_i = (y, x, w/u)_i$ // w/u_i : sequence completion weight //
  end for
  return $\{(y, x, u)\}$
end function
list = []
while size(list)$< n$ **do**
  for all trajectories z^j of length k **do**:
    list.append(get_examples($e^j, \{f_j\}, \{g_j\}, k$))
  end for
end while
$\phi =$ learn(list)
return ϕ

times τ_i are recorded. If no non-matching evidence exists for a variable, $\tau_i = \infty$. Then, instead of sampling transition times from exponential distributions, the times for each X are sampled from truncated exponentials: $t_1 \sim t_0 + q_{x|u} e^{-q_{x|u} t} / (1 - e^{-q_{x|u} \tau})$.

By sampling from truncated exponentials, Fan et al. incur importance sample downweights on their samples. Recall that Equation 2.1 is broken into three components. The weights f_i^*/g_i are decomposed likewise: (1) a downweight for the variable x that transitions, according to the ratio of the exponential to the truncated exponential: $(1 - e^{-q_{x|u}\tau})$, (2) a downweight corresponding to a lookahead point-estimate of Θ_u assuming no other variables change until the evidence (we leave this unmodified in our implementation), and (3) a downweight for each resting variable x_i given by the ratio of not sampling a transition in time t from an exponential and a truncated exponential: $(1 - e^{-q_{x_i|u}\tau_i})/(1 - e^{-q_{x_i|u}(\tau_i - t_1)})$. Finally, the product of all interval downweights provides the trajectory importance sample weight.

While ensuring the validity of each sample, the proposal distribution g in Fan et al. (2010) is non-optimal for sampling complete trajectories. Figure 5.2 shows that stochastic downweighting of some particles occurs at every evidence point, increasing the variance of importance sampling weights each time.

Rejection-Based Importance Sampling in CTBNs

Before we can extend our methodology to CTBNs, we need to show the equivalence of a fixed continuous-time trajectory and the discrete sequences described in Section 5.2. In particular, the rejection-based importance sampling method requires that the number of intervals k must be fixed, while CTBNs produce trajectories

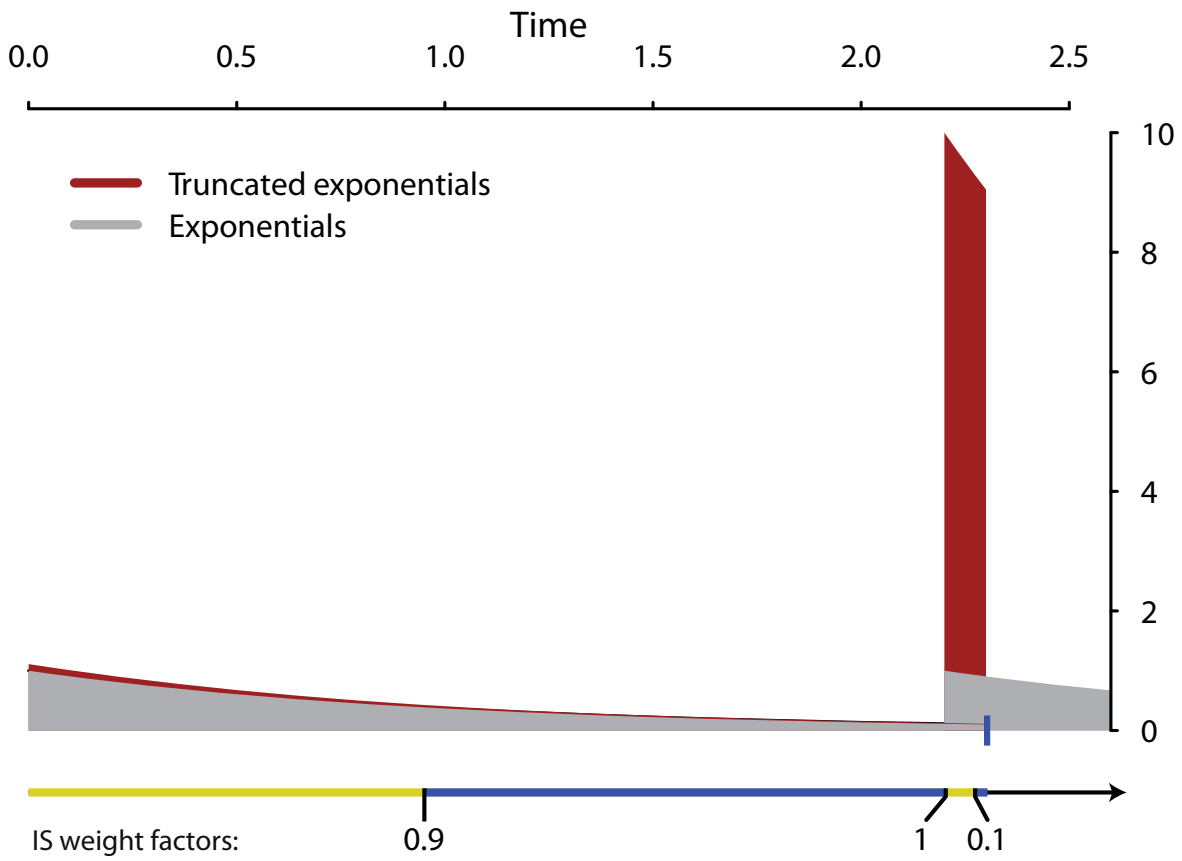

Figure 5.2: Sampling from g results in stochastic downweights. A sample trajectory z, given evidence of "blue" (tick at $t = 2.3$), is shown at bottom, with colors showing the sequence state {yellow, blue, yellow, blue}. The sampled truncated exponentials result in weights equal to the ratio of densities f^*/g (gray/maroon). In this trajectory, the weight is the product of interval weights: (0.9)(1)(0.1). Some sample trajectories pass the evidence with full weight and some do not, resulting in a weight variance factor per evidence point.

with varying numbers of intervals (in fact, the number is unbounded). Nevertheless, for any set of trajectories, we can define ϵ-width intervals small enough that at most one transition occurs per interval and that such transitions occur at the end of the interval. Then for any set of trajectories over the duration $[t_{\text{start}}, t_{\text{end}})$, we set $k = (t_{\text{end}} - t_{\text{start}})/\epsilon$. Using the memoryless property of exponential distributions, it is straightforward to show that the density of a single, one-transition interval is equal to the density of the product of a sequence of ϵ-width, zero-transition intervals and one ϵ-width, one-transition interval. This equivalence between ϵ-width sequences and CTBN trajectories allows us to define CTBNs in relation to the analysis from Section 5.2. In practice, it is simpler to use the CTBN sampling framework so that each interval is of appreciable size. We denote the evidence e as a sequence of tuples of type (state, start time, duration): $(e_i, t_{i,0}, \tau_i)$ allowing for point evidence with zero duration, $\tau_i = 0$, and $\mathbb{1}[e, z]$ checks to see if z agrees with each e_i throughout the duration.

Unlike the discrete-time case where we can enumerate the states z_i, in the continuous-time case the calculation of w_{gh} can be time-consuming because of the normalizing integrals $c_i[g_i, a_i]$. From Equation 5.1, we have, omitting the conditioning:

$$\frac{g_i(z_i)}{h_i(z_i)} = \frac{\int_{\Omega_i} a_i(\zeta) g_i(\zeta) d\zeta}{a_i(z_i)}.$$

For appropriate α and optimal acceptance a^*, we get:

$$\frac{g_i(z_i)}{h_i^*(z_i)} = \frac{\int_{\Omega_i} a_i^*(\zeta) g_i(\zeta) d\zeta}{a_i^*(z_i)} = \frac{\int_{\Omega_i} \alpha^{-1} f_i^*(\zeta) d\zeta}{f_i^*(z_i)(\alpha g_i(z_i))^{-1}} = \frac{1}{w_{i;f^*g}}.$$

For appropriate α and acceptance a_i, we get:

$$\frac{g_i(z_i)}{h_i(z_i)} = \frac{\int_{\Omega_i} a_i(\zeta) g_i(\zeta) d\zeta}{a_i(z_i)} \approx \frac{1}{\alpha a_i(z_i)} = \frac{1 - \phi_i(\cdot)}{\phi_i(\cdot)}. \tag{5.5}$$

The approximation here is that the learned acceptance probability produces a proper conditional probability distribution for each situation. While not guaranteed, the classifier mimics the data distribution, which is drawn from a valid probability distribution. Thus for appropriate, *e.g.*, non-parametric, classifiers and ample data, the approximation error tends to be small as we demonstrate empirically.

For arbitrary α and acceptance a_i, we are left to compute the integral in Equation 5.5. Approaches could include (1) approximating it with samples from g_i, or (2) constraining the classifier to output acceptance probabilities such that the product $g_i(z_i)a_i(z_i)$ can be calculated in closed form. In our experiments, we follow Equation 5.5 and verify the approximation does not introduce significant bias, see, *e.g.*, Figure 5.3 (bottom).

Several properties of CTBNs make our learning framework appealing. First, CTBNs possess the Markov property; namely, the next state is independent of previous states given the current one. Second, CTBNs are homogeneous processes, so the model rate parameters are shared across all intervals. We leverage these facts when learning each acceptance probability a_i. The Markov property simplifies the learned probability of acceptance $\phi_i(z_i|z_{(i-1)\leftarrow 1}, e, f, g)$ to $\phi_i(z_i|z_{(i-1)}, e, f, g)$. Homogeneity simplifies the learning process because, if $z_{(i-1)} = z_{(j-1)}$ and $t_{i,0} = t_{j,0}$ for $j \neq i$, then $\phi_i(z_i|z_{(i-1)}, e, f, g) = \phi_i(z_j|z_{(j-1)}, e, f, g)$. The degeneracy of these two cases indicates that the probability of acceptance is a function of the situation and independent of the interval index, so a single classifier can be learned in place of k classifiers.

5.4 Experiments

We compare our learning-based rejection method (setting $\alpha = 2$) with the truncated exponential sampler for modeling CTBNs (Fan et al., 2010). We learn a logistic regression (LR) model using online, stochastic gradient descent for each CTBN state. An LR data example is $(y, \mathbf{x}, w)$, where y is one of $\{\text{accept}, \text{reject}\}$, $\mathbf{x}$ is a set of features, and w is the sequence completion weight. For our experiments we use per-variable features encoding the state (as indicator variables), the time from the current time to next evidence, the time from the proposed sample to next evidence, and the time from the proposed sample to next matching evidence. The times are mapped to intervals $[0, 1]$ by using a $e^{-t/\lambda}$ transformation, with $\lambda = \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ to capture effects at different time scales. Because of the high variance in weights for samples of full sequence completions, we instead choose a local weight: the weight of the sequence through the next $m = 10$ evidence times. This biases the learner to have low variance weights within the local window, but it does not bias the proposal h.

We analyze the performance of the rejection-based sampler by inspection of learned transition probability densities and the effective sample size (ESS) (Kong et al., 1994). ESS is an indicator of the quality of the samples, and a larger value is better: $\text{ESS} = 1/(\sum_{i=1}^{n}(W^i)^2)$, where $W^i = w^i / \sum_{j=1}^{n} w^j$. We test our method in several models: one-, two- and three- variable, strong-cycle binary-state CTBNs, and the partially-binary,

Table 5.1: Table of the geometric mean of effective sample size (ESS) over 100 sequences, each with 100 observations; ESS is per 10^5 samples. The proposal h was learned with 1000 sequences.

Model	Fan et al. (g)	Rejection SIS (h)
Strong cycle, n=1	690	6400
Strong cycle, n=2	19000	35000
Strong cycle, n=3	960	5800
Drug	29	170

8-variable drug model presented in the original CTBN paper (Nodelman, 2007). The "strong-cycle" models encode an intensity path for particular joint states by shifting bit registers and adding and filling in an empty register with a 0 or 1 as in the following example. For the 3-variable strong cycle, intensities involved in the path $000 \rightarrow 001 \rightarrow 011 \rightarrow 111 \rightarrow 110 \rightarrow 100 \rightarrow 000$ are 1, and intensities are 0.1 for all other transitions. We generate sequences from ground truth models and censor each to retain only 100 point evidences with times t_i drawn randomly, uniformly over the duration [0,20). We provide the active state, intensities, point evidence times, variables and states in the form of features to the classifier.

Figure 5.3 (top) illustrates the ability of h to mimic f^*, the target distribution, in a one-node binary-state CTBN with matching evidence at $t = 5$. The Fan et al. proposal g is chosen to match the target density in the absence of evidence f. However, when approaching evidence (at $t = 5$), the probability of a transition given evidence goes to 0 as the next transition must also occur before $t = 5$ to be a viable sequence. Only f^* and h exhibit this behavior. Figure 5.3 (bottom) shows the density approximations after weighting the samples, given a trajectory with evidence at $t = 5$ and 19 evidence points after $t = 20$. Each method recovers the target distribution, but h does so more precisely than g, given a fixed number of samples (one million). As a proxy for the extra work required to sample from h, the proposal acceptance rate was measured to be 45 percent.

Table 5.1 shows that the learned, rejection-based proposal h outperforms the other CTBN importance sampler g across all 4 models, resulting in an ESS 2 to 10 times larger. Generally as the number of variables increases, the ESS decreases because of the increasing mismatch between f^* and g. With an average ESS of only 29 in an 8-variable model, as we increase model sizes, we expect that g would fail to produce a meaningful sample distribution more quickly than h would.

To illustrate further, using the drug model and 10 evidence points, Figure 5.4 shows that the weight distribution from h is narrower than that from g on the log scale. The interpretation is that a larger fraction of examples from h contribute substantially to the total weight, resulting in a lower variance sample distribution. For example, any sample with weight below e^{-10} has negligible relative weight and does not substantially affect the sample distribution. There are many fewer such samples generated from h than from g.

5.5 Discussion

Continued investigations are warranted, and we discuss several possible extensions: (1) the use of non-parametric learning algorithms, (2) a procedure for expanding the local weight approximations to the full sequence, and (3) an iterative procedure to learn the acceptance probability and construct a new proposal g' not requiring rejection sampling. First, in our experiments we used logistic regression models; however, we know that there is a correct outcome for every "situation", so the function we wish to learn is distribution-free. Non-parametric learning algorithms are appropriate for such problems, although their relative lack of scalability to large dimensions could be problematic. Ones that take into account the form of the normalizing

functionals $c_i[g_i a_i]$ should also be investigated. Second, the generation of examples with weights reflecting the expected weight ratio is intractable for the full sequence. In our experiments we show a good approximation using weights based on a window of m evidence steps. We believe we can iteratively extend the window size to decrease the approximation error. Finally, if the initial surrogate distribution g_i is far from f_i^*, the rejection rate $r = (\alpha - 1)/\alpha = \max f_i^*(\cdot)/g_i(\cdot)$ must be large to recover f_i^* from g_i. Instead, a closed-form function approximating h_i can be used to learn a new rejection-based proposal h_i'. Akin to works in adaptive importance sampling, *e.g.*, (Cheng & Druzdzel, 2000), this iterative procedure would generate an improving sequence of closed-form sampling distributions while lowering the rate of rejection.

5.6 Summary

Our work has demonstrated that machine learning can be used to improve sequential importance sampling via a rejection sampling framework. First, we showed that the proposal and target distributions are related by an expected weight ratio. Then, the weight ratio can be estimated by the probabilistic output of a binary classifier learned from weighted, local importance samples. We extended the algorithm to CTBNs, where we found experimentally that using our learning algorithm produces a sampling distribution closer to the target and generates more effective samples.

Despite this, our algorithm will have trouble with EHR-sized inference problems, so in the next chapter we consider point processes which sidestep the CTBN inference problem altogether. We maintain that CTBNs richly model uncertainty and are often preferable in cases where inference is possible. The next chapter highlights the differences between CTBNs and point processes in detail.

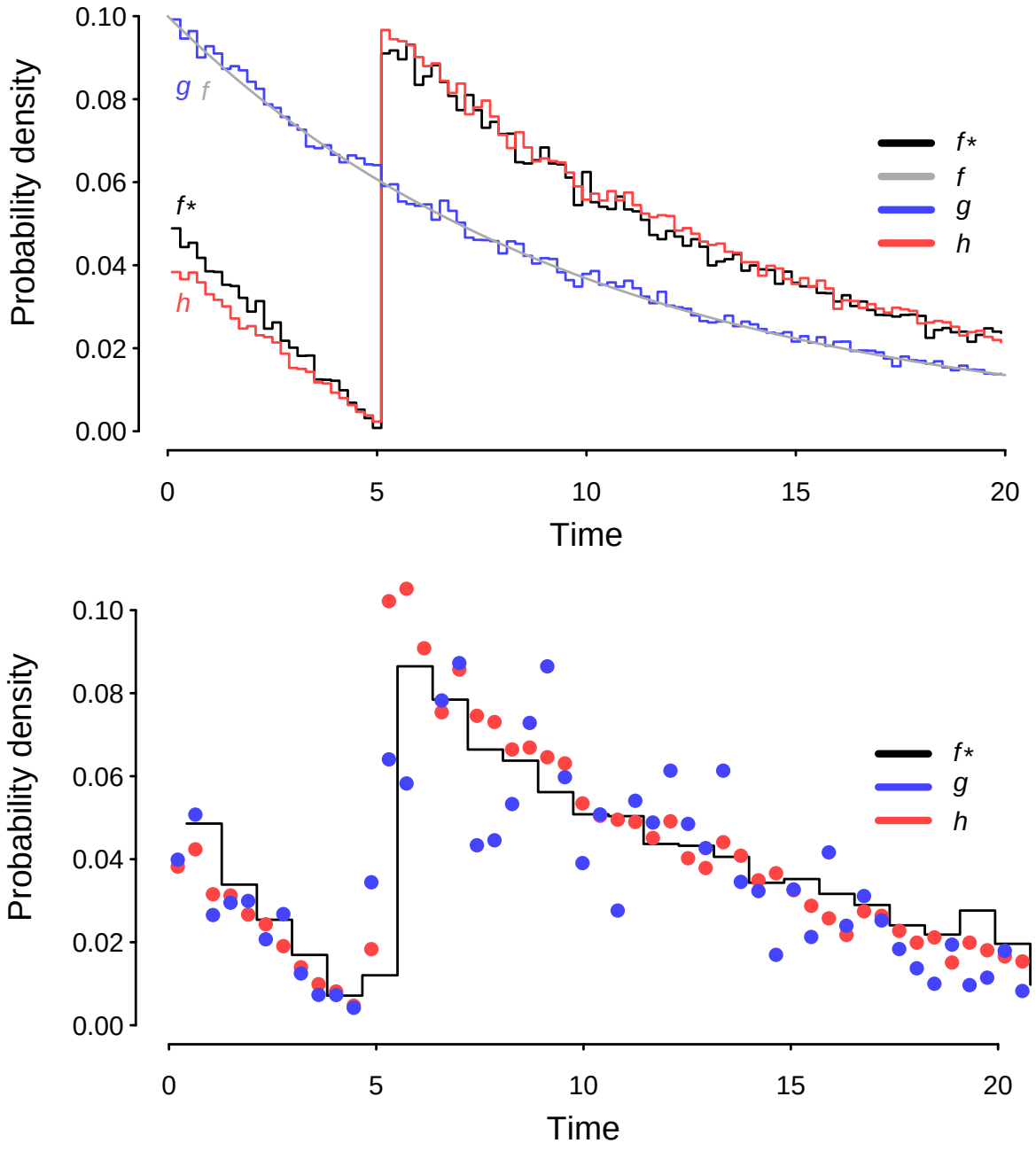

Figure 5.3: Approximate transition densities (top) of f^* (target), f (target without evidence), g (surrogate), and h (learned rejection surrogate) in a one-variable, binary-state CTBN with uniform transition rates of 0.1 and matching evidence at t=5. The learned distribution h closely mimics f^*, the target distribution with evidence, while g was constructed to mimic f (exactly, in this situation). All methods recover the weighted transition densities (bottom); for 20 evidence points with one at t=5 and 19 after t=20, h recovers the target distribution more precisely than g per 10^6 samples.

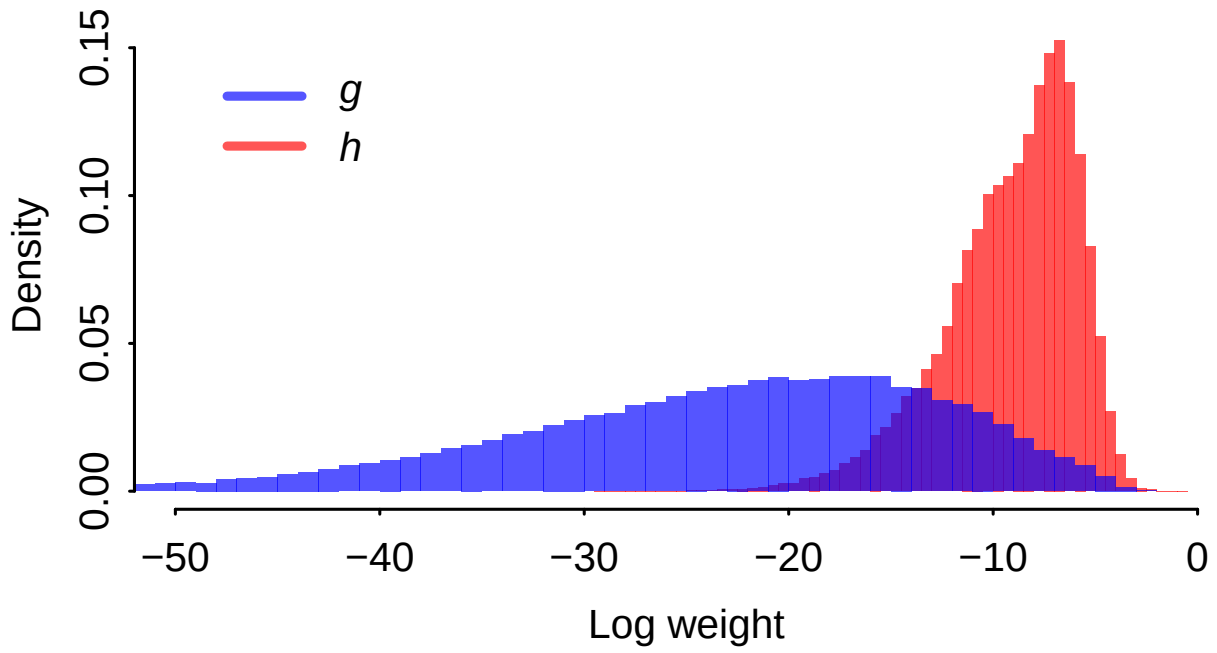

Figure 5.4: Distribution of log weights. For sample completions of a trajectory with 10 evidence points in the drug model, the distribution of log weights using h is much narrower than the distribution of log weights using g (bottom).

6 Learning Multiplicative Forests for Point Processes and Event Prediction from Electronic Health Records

Overview

Motivated by the challenges in CTBN inference we turn now to point process, an alternative model that models event data arriving at semi-irregular intervals. We extend our CTBN forest idea to build multiplicative-forest point processes (MFPPs), which learn the rate of future events based on an event history. MFPPs join theory in partition-based continuous-time Bayesian networks and piecewise-continuous conditional intensity models. We analyze the advantages of using MFPPs over previous methods and show that on synthetic and real EHR forecasting of heart attacks, MFPPs outperform earlier methods and augment off-the-shelf machine learning algorithms. This work is based on published work in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2013).

6.1 Introduction

EHRs record information about individuals who have regular check-ups interspersed with hospitalizations and medical emergencies. These sequences of semi-irregular events can be considered as timelines. However, the majority of models incorporating time use a time-series data representation, where data are assumed to arrive at regular intervals. Irregular arrivals of events violate this assumption and lead to missing data and/or aggregation, resulting in a loss of information. Experimentally, such methods have been shown to underperform analogous continuous-time models (Nodelman et al., 2003).

To address the irregularity of medical event arrivals, we develop a continuous-time model: multiplicative-forest point processes (MFPPs). Unlike CTBNs, which model event durations, MFPPs model the rate of event occurrences. Futhermore, they make the assume that they are dependent on an event history in a piecewise-constant manner. For example, the event of aspirin consumption (or lack thereof) may affect the rate of myocardial infarction, or heart attack, which in turn affects the rate of thrombolytic therapy administration. Our goal is to learn a model that identifies such associations from data.

MFPPs build on previous work in piecewise-constant conditional intensity models (PCIMs) using ideas from multiplicative-forest continuous-time Bayesian networks (mfCTBNs) (Gunawardana et al., 2011; Weiss et al., 2012c). MFPPs extends the regression tree structure of PCIMs to regression forests. Unlike most forest learning algorithms, which minimize a classification loss through functional gradient ascent or ensembling, MFPPs are based on a multiplicative-forest technique developed in CTBNs. Here, a multiplicative assumption for combining regression tree values leads to optimal marginal log likelihood updates with changes in forest structure. The multiplicative representation allows MFPPs to concisely represent composite rates, yet also to have the flexibility to model rates with complicated dependencies. As the multiplicative forest model leads to representational and computational gains in mfCTBNs, we show that similar gains can be achieved in the point process domain. We conduct experiments to test two main hypotheses. First, we test for improvements in learning MFPPs over PCIMs, validating the usefulness of the multiplicative-forest concept. Second, we assess the ability of MFPPs to classify individuals for myocardial infarction from EHR data, compared to PCIMs and off-the-shelf machine learning algorithms.

Specifically we address two modeling scenarios for forecasting: *ex ante* (meaning "from the past") forecasting and supervised forecasting. An *ex ante* forecast is the traditional type of forecasting and occurs if no labels are available in the forecast region. An example of *ex ante* forecasting is the prediction of future disease onset from the present day forwards. Acquiring labels from the future is not possible, and labels from the past may introduce bias through a cohort effect. However, in some cases, labels may be used, and we call such forecasts "supervised". An example of supervised forecasting is the retrospective cohort study to predict the class of unlabeled examples as well as to identify risk factors leading to disease. The application of continuous-time models to the forecasting case is straightforward. When labels are available, however, we choose to apply MFPPs in a cascade learning framework, where the MFPP predictions contribute as features to supervised learning models.

In Section 6.2, we discuss point processes and contrast them from continuous-time Bayesian networks (CTBNs) noting their matching likelihood formulations given somewhat different problem setups. We show that multiplicative forest methods can be extended to point processes. We also introduce the problem of predicting myocardial infarction, discuss the various approaches to answering medical queries, and introduce our method of analysis. In Section 6.3, we present results on synthetic timelines and real health records data and show that MFPPs outperform PCIMs on these tasks, and that the timeline analysis approach outperforms other standard machine learning approaches to the problem. We conclude in Section 6.4.

6.2 Point Processes

We note that with this assumption the likelihood formulation becomes identical to the one used in continuous-time Bayesian networks (CTBNs). The shared likelihood formula lets us apply a recent advance in learning CTBNs: the use of multiplicative forests. Multiplicative forests produce intensities by taking the product of the regression values in active leaves. For example, a multiplicative forest equivalent to the tree described above is shown in Figure 6.1 (right). These models were shown to have large empirical gains for parameter and structure learning similar to those seen in the transition from tree models to random forests or boosted trees (Weiss et al., 2012c). Our first goal is to show that a similar learning framework can be applied to point processes. We describe the model in fuller detail below.

Piecewise-Continuous Conditional Intensity Models (PCIMs)

Recall from Chapter 2 the form of the point process: given a finite set of event types $l \in \mathcal{L}$, an event sequence or trajectory x is an ordered set of {time, event} pairs $(t, l)_{i=1}^{n}$. Given a history h of event, the likelihood of the trajectory given the CIM θ is:

$$p(x|\theta) = \prod_{l \in \mathcal{L}} \prod_{i=1}^{n} \lambda_l(t_i|h_i, \theta)^{\mathbb{1}(l=l_i)} e^{\int_{-\infty}^{t} \lambda_l(\tau|x,\theta)d\tau}$$

PCIMs introduce the assumption that the intensity functions are constant over intervals. As described in (Gunawardana et al., 2011), let Σ_l be a set of discrete states so that we obtain the set of parameters λ_{ls} for $s \in \Sigma_l$. The active state s is determined by a mapping $\sigma_l(t, x)$ from time and trajectory to s. Let S_l hold the pair $(\Sigma_l, \sigma_l(t, x))$ and let $S = \{S_l\}_{l \in \mathcal{L}}$. Then the PCIM likelihood simplifies to:

$$p(x|S, \theta) = \prod_{l \in \mathcal{L}} \prod_{s \in \Sigma_l} \lambda_{ls}^{M_{ls}(x)} e^{-\lambda_{ls} T_{ls}(x)}, \tag{6.1}$$

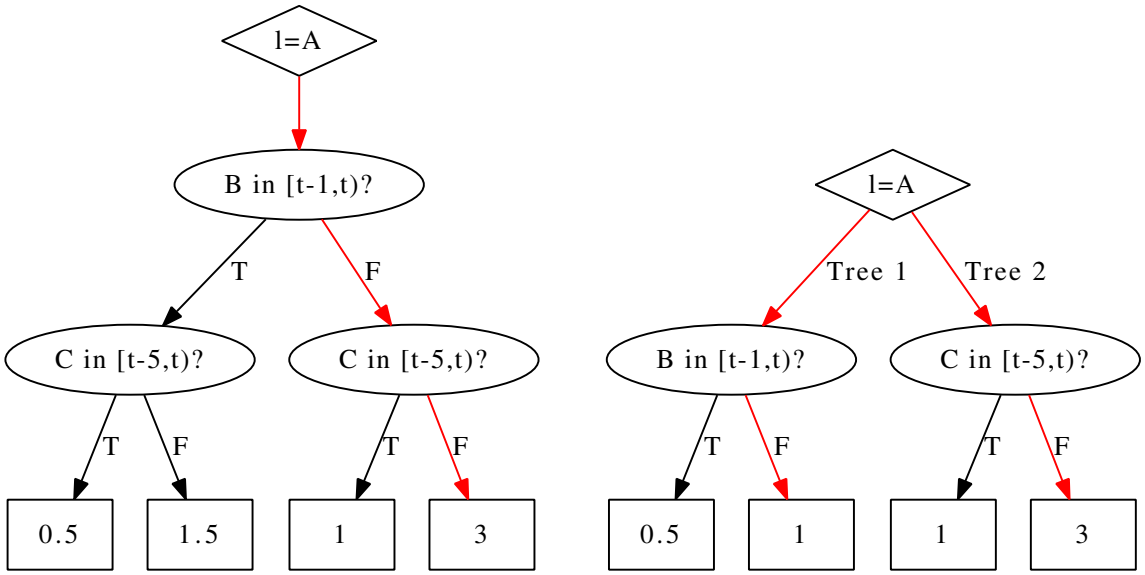

Figure 6.1: A piecewise-constant conditional intensity tree for determining the rate of event type A (left). An equivalent multiplicative intensity forest (right). An example of active paths are shown in red. The active path in the tree corresponds to the intersection of active paths in the forest, and the output intensity is the same ($3 = 1 \times 3$).

where $M_{ls}(x)$ is the count of events of type l while s is active in trajectory x, and $T_{ls}(x)$ is the total duration that s, for event type l, is active.

Continuous-Time Bayesian Networks (CTBNs)

For clarity, we restate the CTBN likelihood formulation here. A trajectory, or a timeline, is broken down into independent intervals of fixed state. For each interval $[t_0, t_{\text{end}})$, the duration $t = t_{\text{end}} - t_0$ passes and a variable x transitions at t_{end} from state x^j to x^k. All other variables $x_i \neq x$ rest during this interval in their active states x_i'. Then, the interval density is given by:

$$\underbrace{\lambda_{x^j|u} e^{-\lambda_{x^j|u} t}}_{x \text{ transitions}} \underbrace{\Theta_{x^j x^k|u}}_{\text{to state } x^k} \underbrace{\prod_{x_i' : x_i \neq x} e^{-\lambda_{x_i'|u} t}}_{\text{while } x_i\text{'s rest}}$$

The trajectory likelihood is given by the product of intervals:

$$\prod_{x \in \mathcal{X}} \prod_{x^j \in x} \prod_{u \in U_x} \lambda_{x^j|u}^{M_{x^j|u}} e^{-\lambda_{x^j|u} T_{x^j|u}} \prod_{x^k \neq x^j} \Theta_{x^j x^k|u}^{M_{x^j x^k|u}} \tag{6.2}$$

where the $M_{x^j|u}$ (and $M_{x^j x^k|u}$) are the numbers of transitions out of state x^j (to state x^k), and where the $T_{x^j|u}$ are the amounts of time spent in x^j given parents settings u. Defining rate parameter $\lambda_{x^i x^j|u} = \lambda_{x^i|u} \Theta_{x^i x^j|u}$ and set element $p = x^j \times u$ (as in (Weiss et al., 2012c)), Equation 6.2 can be rewritten as:

$$\prod_{x \in \mathcal{X}} \prod_{x' \in x} \prod_{p} \lambda_{x'|p}^{M_{x'|p}} e^{-\lambda_{x'|p} T_p} \tag{6.3}$$

Table 6.1: Contrasting piecewise-constant continuous intensity models (PCIMs) and multiplicative-forest continuous-time Bayesian networks (mfCTBNs). Key similarities are highlighted in blue.

	PCIM	mfCTBN
Model of:	event sequence	persistent state
Intensities	piecewise-constant	network-dependent constant
Dependence	event history	joint state (Markovian)
Labels	event types	variables
Emissions	events	states (x', 1 of s_i)
Structure	regression tree	multiplicative forest
Evidence	events	(partial) observations of states
Likelihood	$\prod_l \prod_s \lambda_{ls}^{M_{ls}} e^{-\lambda_{ls} T_{ls}}$	$\prod_{x'} \prod_p \lambda_{x'\|p}^{M_{x'\|p}} e^{-\lambda_{x'\|p} T_p}$

Note how the form of the likelihood in Equation 6.1 is identical to Equation 6.3.

Contrasting PCIMs and CTBNs

Despite the similarity in form, PCIMs and CTBNs model distinctly different types of continuous-time processes. Table 6.1 contrasts the two models. The primary difference is that, unlike point processes, CTBNs model a persistent, joint state over time. That is, a CTBN provides a distribution over the joint state for any time t. Additionally, CTBN variables must possess a 1-of-s_i state representation for $s_i > 1$ whereas point processes typically assume non-complementary event types. Furthermore, in CTBNs, observations are typically not of changes in state at particular times but instead probes of the state at a time point or interval. With persistent states, CTBNs can be used to answer interpolative queries, whereas CIMs are designed specifically for forecasting. Another notable difference is that CTBNs are Markovian: the intensities are determined entirely by the current state of the system. While more restrictive, this assumption allows for variational and MCMC methods to be applied. On the other hand, PCIMs lend themselves to forecasting because the potentially prohibitive inference about the persistent state that CTBNs require is no longer necessary. This is because the rate of event occurrences depends on the event history instead of the current state.

Multiplicative-Forest Point Processes (MFPPs)

The similar likelihood forms allow us to extend the multiplicative-forest concept (Weiss et al., 2012c) to PCIMs. Following (Gunawardana et al., 2011), we define the state Σ_l and mapping $\sigma_l(t, x)$ according to regression trees. Let $\mathcal{B}_l$ be the set of basis state functions $f(t, x)$ that maps to a basis state set Σ_f, akin to $\sigma(t, x)$ that maps to a single element s. As in (Weiss et al., 2012c), we can view the basis functions as set partitions of the space over $\Sigma = \Sigma_{l_1} \times \Sigma_{l_2} \times \ldots \Sigma_{l_{|\mathcal{L}|}}$. Each interior node in the regression tree is associated with a basis function f. Each leaf holds a non-negative real value: the intensity. Thus one path ρ through the regression tree for event type l corresponds to a recursive subpartition resulting in a set Σ_ρ, and every $(l, s) \in \Sigma_\rho$ corresponds to leaf intensity $\lambda_{l\rho}$, i.e., we set $\lambda_{ls} = \lambda_{l\rho}$. Figure 6.1 shows an example of the active path providing the intensity ($\lambda_{ls} = \lambda_{l\rho} = 3$).

MFPPs replace these trees with random forests. Given that each tree represents a partition, the intersection of trees, *i.e.* a forest, forms a finer partition. The subpartition corresponding to a single intensity is given by

the intersection $\Sigma_\rho = \bigcap_{j=1}^{k} \Sigma_{\rho,j}$ of sets corresponding to the active paths through trees $1 \dots k$. The intensity $\lambda_{l\rho}$ is given by the product of leaf intensities. Figure 6.1 (right) shows an example of the active paths in a tree, producing the forest intensity ($\lambda_{ls} = \lambda_{l\rho} = 1 \times 3$).

MFPPs use the PCIM generative framework. Forecasting is performed by forward sampling or importance sampling to generate an approximation to the distribution at future times. Learning MFPPs is analogous to learning mfCTBNs. A tree is learned iteratively by replacing a leaf with a branch with corresponding leaves. As in forest CTBNs, MFPPs have (1) a closed form marginal log likelihood update and (2) a simple maximum likelihood calculation for modification proposals. The intensities for the modification are the ratios between observed (M_{ls}) divided by expected ($\lambda_{ls} T_{ls}$) number of events prior to modification and while Σ_ρ is active. These two properties together provide the best greedy update to the forest model.

The use of multiplicative forest point processes has several advantages over previous methods.

- Compared to trees, forest models can represent more intensities per parameter, which is equal to the number of leaves in the model. For example, if a ground truth model has k stumps, that is, k single-split binary trees, then the forest can represent the model with $2k$ parameters. An equivalent tree would require 2^k parameters. This example arises whenever two risk factors are independent, i.e., their risks multiply.

- While forests can represent these independences when needed, they also can represent non-linear processes by increasing the depth of the tree beyond one. This advantage was established in previous work comparing trees to Poisson Networks (Gunawardana et al., 2011; Rajaram et al., 2005), and forests possess advantages of both approaches.

- Unlike most forest models, multiplicative-forest trees may be learned in an order that is neither sequential nor simultaneous. The forest appends a stump to the end of its tree list when that modification improves the marginal likelihood the most. Otherwise it increases the depth of one tree. The data determines which expansion is selected.

- Multiplicative forests in CTBNs are restricted to learning from the current state (the Markovian assumption), whereas MFPPs learn from a basis set over some combination of the event history, deterministic, and constant features.

- Compared to the application of supervised classification methods to temporal data, the point process model identifies patterns of event sequences over time and uses them for forecasting. Figure 6.2 shows an example of the supervised forecasting setup. In this case, it may be harder to predict event B without using recurrent patterns of event sequences.

We hypothesize that these advantages will result in improved performance at forecasting, particularly in domains where risk factors are independent. As many established risk factors for cardiovascular disease are believed to contribute to the overall risk independently, we believe that MFPPs should outperform tree methods at this task. Because of their facility in modeling irregular series of events, we also believe that MFPPs should also outperform off-the-shelf machine learning methods.

Related Work

A rich literature exists on point processes focusing predominantly on spatial forecasting. In spatial domains, the point process is the temporal component of a model used to predict spatiotemporal patterns in data.

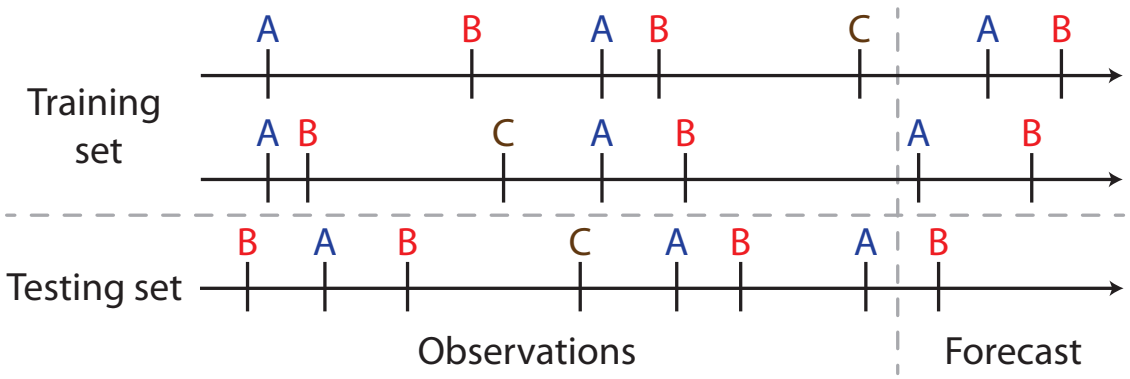

Figure 6.2: Supervised forecasting. Labels are provided by the binary classification outcome: whether at least one event occurs in the forecasted region.

The analysis of multivariate, spatial point processes is related to our work in its attempt to characterize the joint behavior of variables, for example, using Ripley's K function test for spatial homogeneity (Ripley, 1976). However, these methods do not learn dependency structures among variables; instead they seek to characterize cross-correlations observed in data. Generalized linear models for simple point processes are more closely related to our work. Here, a linear assumption for the intensity function is made, seen for example in Poisson networks (Rajaram et al., 2005). PCIMs adopt a non-parametric approach and was shown to substantially improve upon previous methods in terms of model accuracy and learning time (Gunawardana et al., 2011). Our method builds on upon the PCIM framework.

Risk assessment for cardiovascular disease is also well studied. The primary outcome of most studies is the identification of one or a few risk factors and the quantification of the attributable risk. Our task is slightly different; we seek to predict from data the onset of future myocardial infarctions. The prediction task is closely related to risk stratification. For cardiovascular disease, the Framingham Heart Study is the landmark study for risk assessment (Wilson et al., 1998). They provide a 10-year risk of cardiovascular disease based on age, cholesterol (total and HDL), smoking status, and blood pressure. A number of studies have been since conducted purporting significant improvements over the Framingham Risk Score using different models or by collecting additional information (Tzoulaki et al., 2009). In particular, the use of EHR data to predict heart attacks was previously addressed in Weiss et al. (2012b). However, in that work the temporal dependence of the outcome and its predictors was strictly logical and limited the success of their approach. We seek to show that, compared to standard approaches learning from features segmented in time, a point process naturally models timeline data and results in improved risk prediction.

6.3 Experiments

We evaluate MFPPs in two experiments. The first uses a model of myocardial infarction and stroke, and the goal is to learn MFPPs to recover the ground truth model from sampled data. The second is an evaluation of MFPPs in predicting myocardial infarction diagnoses from real EHR data.

Model Experiment: Myocardial Infarction and Stroke

We introduce a ground truth PCIM model of myocardial infarction and stroke. The dependency structure of the model is shown in Figure 6.3. To compare MFPPs with PCIMs, we sample k trajectories from time 0 to 80

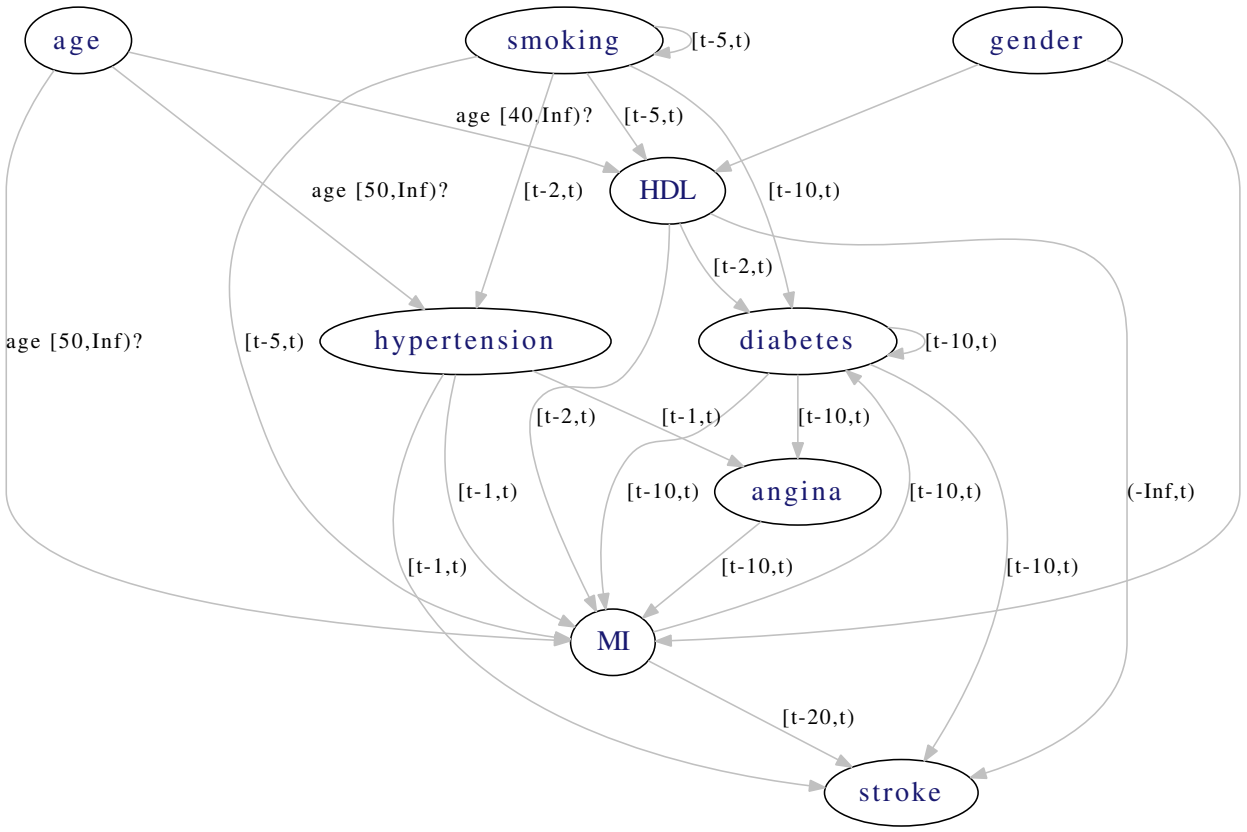

Figure 6.3: Ground truth dependency structure of heart attack and stroke model. Labels on the edges determine the active duration of the dependency. Omitted in the graph is the age dependency for all non-deterministic nodes if the subject is older than 18.

for $k = \{50, 100, 500, 1000, 5000, 10000\}$. We train each model with these samples and calculate the average log likelihood on a testing set of 1000 sampled trajectories. Each model used a BIC penalty to determine when to terminate learning. For features, we constructed a feature generator that uniformly at random selects an event type trigger and an active duration of one of $\{t-1, t-5, t-10, t-20, t-50\}$ to t. Note that the feature durations do not have a direct overlap with the dependency intervals shown in Figure 6.3. Our goal was to show that, even without being able to recover the exact ground truth model, we could get close with surrogate features. MFPPs were allowed to learn up to 10 trees each with 10 splitting features; PCIMs were allowed 1 tree with 100 splitting features. We also performed a two-tailed paired t-test to test for significant differences in MFPP and PCIM log likelihood. We ran each algorithm 250 times for each value of k.

Figure 6.4 shows the average log likelihood results. Both MFPPs and PCIMs appear to converge to close to the ground truth model with increasing training set sizes. The lack of complete convergence is likely due to the mismatch in ground truth dependencies and the features available for learning. Error bars indicating the empirical 95 percent confidence intervals are also shown for MFPP. Similar error bars were observed for the ground truth and PCIM models but were omitted for clarity. The width of the interval is due to the variance in testing set log likelihoods. If we look at level average log likelihood lines in Figure 6.4, we observe that we only need a fraction of the data to learn a MFPP model equally good as the PCIM model. Both models completed all runs in under 15 minutes each.

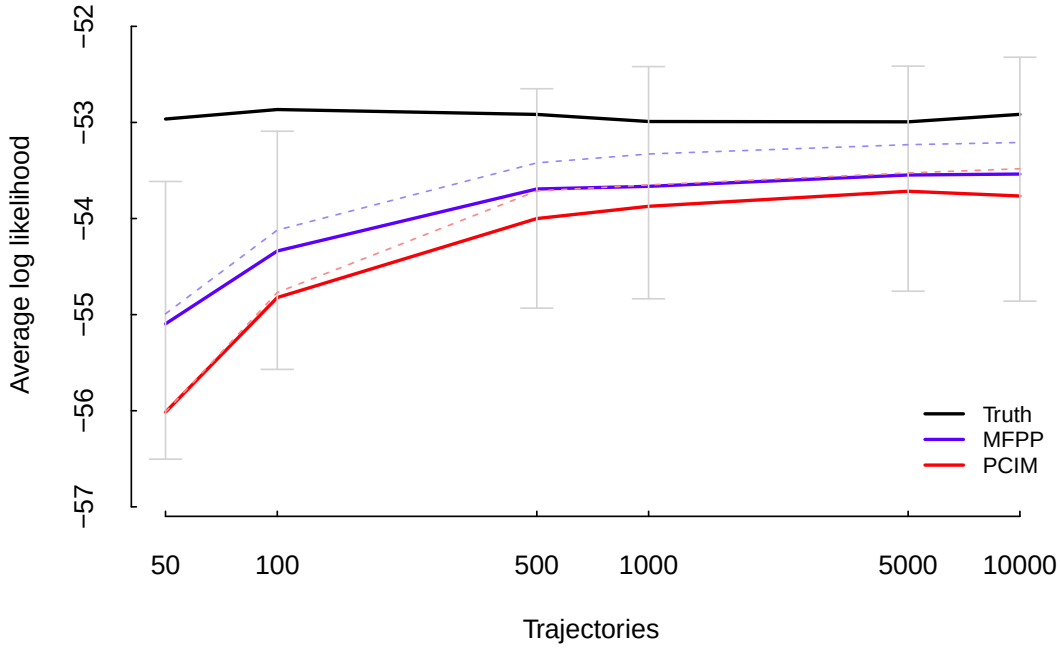

Figure 6.4: Average log likelihoods for the {ground truth, MFPP, PCIM} model by the number of training set trajectories. Error bars in gray indicate the 95 percent confidence interval (omitted for the ground truth and PCIM models). Paired t-tests comparing MFPPs and PCIMs were significant at a p-value of 1-e20. Dotted lines show the likelihoods when ground truth features were made available to the models.

We used a two-sided paired t-test to test for significant differences in the average log likelihood. For all numbers of trajectories k, the p-value was smaller than 1e-20. We conclude that the MFPP algorithm significantly outperformed the PCIM algorithm at recovering the ground truth model from data of this size.

EHR Prediction: Myocardial Infarction

In this section we describe the experiment on real EHR data. We define the task to be forecasting future onset of myocardial infarction between the years 2005 and 2010 given event data prior to 2005. We propose two forms of this experiment: *ex ante* and supervised forecasting. First, we test the ability of MFPP to forecast events between 2005 and 2010 in all patients given the data leading up to 2005. Figure 6.5 depicts the *ex ante* forecasting setup.

Second, we split our data into training and testing sets to test MFPP in its ability to perform supervised forecasting. In this setup, we provide data between 2005 and 2010 for the training set in addition to all data prior to 2005 for both training and testing sets. We choose to focus on the outcome of whether a subject has at least one myocardial infarction event between the 2005 and 2010. Figure 6.2 shows the supervised forecasting setup.

We use EHR data from the Personalized Medicine Research Project (PMRP) cohort study run at the Marshfield Clinic Research Foundation (McCarty et al., 2005). The Marshfield Clinic has followed a patient population residing in northern Wisconsin and the outlying areas starting in the early 1960s up to the present.

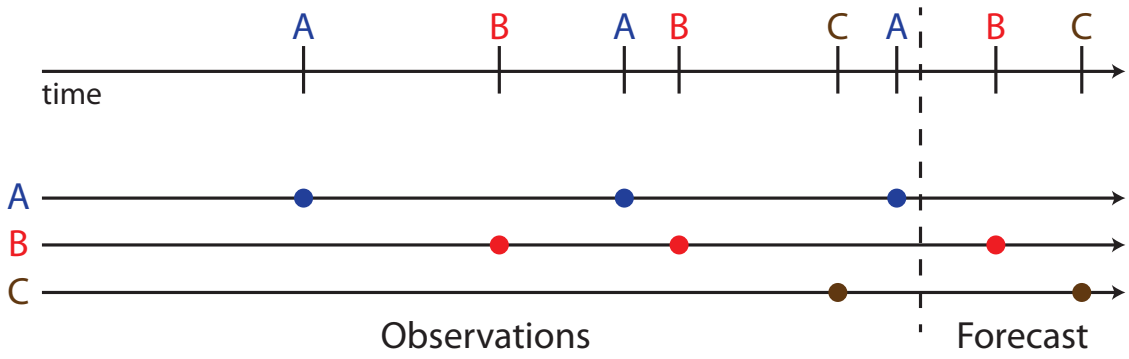

Figure 6.5: *Ex ante* (traditional) forecasting. No labels for any example are available in the forecast region. The goal is to recover the events (B and C) from observations in the past.

From this cohort, we include all subjects with at least one event between 1970 and 2005, and with at least one event after 2010 or a death record after 2005. Filtering with these inclusion criteria resulted in a study population of 15,446, with 428 identified individuals with a myocardial infarction event between 2005 and 2010.

To make learning and inference tractable, we selected additional event types from the EHR corresponding to risk factors identified in the Framingham Heart Study(Wilson et al., 1998): age, date, gender, LDL (critical low, low, normal, high, critical high, abnormal), blood pressure (normal, high), obesity, statin use, diabetes, stroke, angina, and bypass surgery. Because the level of detail specified in EHR event codes is fine, we use the above terms that represent aggregates over the terms in our database, *i.e.*, we map the event codes to one of the coarse terms. For example, an embolism lodged in the basilar artery is one type of stroke, and we code it simply as "stroke". The features we selected produced an event list with over 1.8 million events. As MFPPs require selecting active duration windows to learn, we used durations of size {0.25, 1, 2, 5, 10, 100 (ever)}, with more features focused on the recent past. Our intuition suggests that events occurring in the recent past are more informative than more distant events.

We compare MFPP against two sets of machine learning algorithms based on the experimental setup. For *ex ante* forecasting, we test against PCIMs (Gunawardana et al., 2011) and homogeneous Poisson point processes, which assume independent and constant event rates. We assess their performance using the average log likelihood of the true events in the forecast region and precision-recall curves for our target event of interest: myocardial infarction. For supervised forecasting, we test against random forests and logistic regression (Gunawardana et al., 2011; Breiman, 2001). As MFPP is not an inherently supervised learning algorithm, we also include a random forest learner using features corresponding to the intensity estimates based on the *ex ante* forecasting setup. We call this method MFPP-RF. We use modified bootstrapping to generate non-overlapping training and testing sets, and we train on 80 percent of the entire data. We compare the supervised forecasting methods only in terms of precision-recall due to the non-correspondence of the methods' likelihoods.

We also make a small modification to the MFPP and PCIM learning procedure when learning for modeling myocardial infarction, i.e., rare, events. On each iteration we expand one node in the forest of every event type instead of the forest of a single event type. The reason for this is that low intensity variables contribute less to the likelihood, so choosing the largest change in marginal log likelihood will tend to ignore modeling low intensity variables. By selecting an expansion for every event type each iteration, we ensure a rich modeling of myocardial infarction in the face of high frequency events such as blood pressure measurements

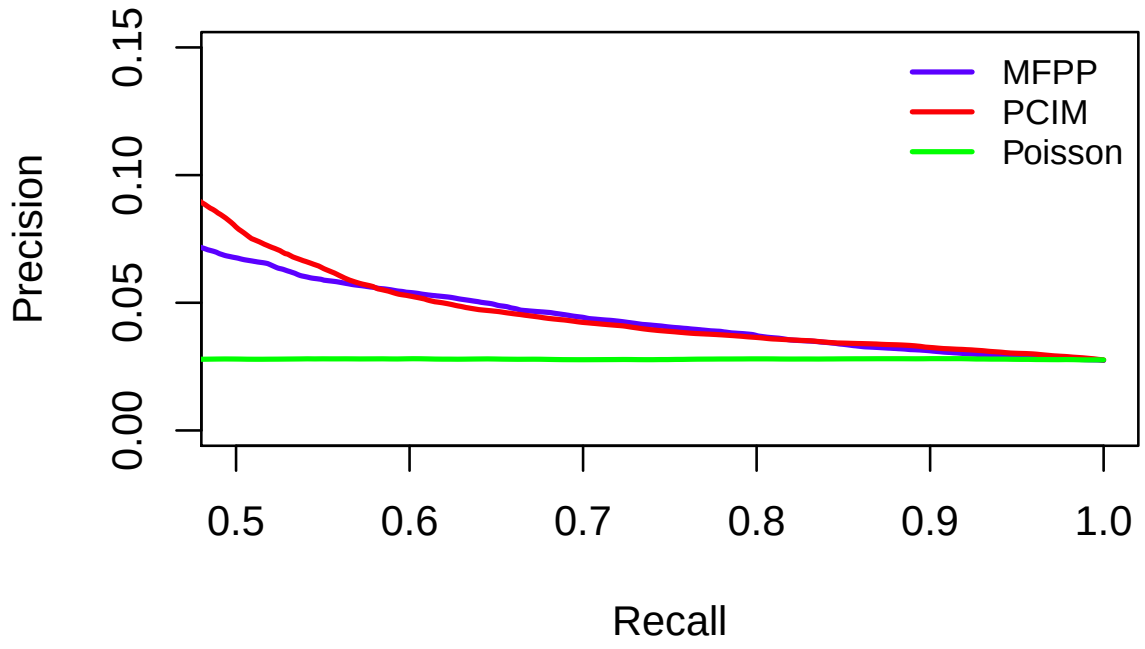

Figure 6.6: Precision-recall curves for *ex ante* forecasting. MFPPs are compared against PCIMs and homogeneous Poisson point processes.

Table 6.2: Log likelihood of {MFPP, PCIM, independent homogeneous Poisson processes} for forecasting patient medical events between 2005 and 2010.

Method	Log likelihood
MFPP	12.1
PCIM	10.3
Poisson	-54.8

and prescription refills. We note that because of the independence of likelihood components for each event type, this type of round-robin expansion is still guaranteed to increase the model likelihood. This statement would not hold, for example, in CTBNs, where a change in a variable intensity may change its latent state distribution, affecting the likelihood of another variable. Finally, for ease of implementation and sampling, we learn trees sequentially and limit the forest size to 40 total splits.

Ex Ante Forecasting Results

Table 6.2 shows the average log likelihood results for *ex ante* forecasting for the MFPP, PCIM and homogeneous Poisson point process models. Both MFPPs and PCIMs perform much better than the baseline homogeneous model. MFPPs outperform PCIMs by a similar margin observed in the synthetic data set.

Figure 6.6 shows the precision-recall curve for predicting a myocardial infarction event between 2005 and

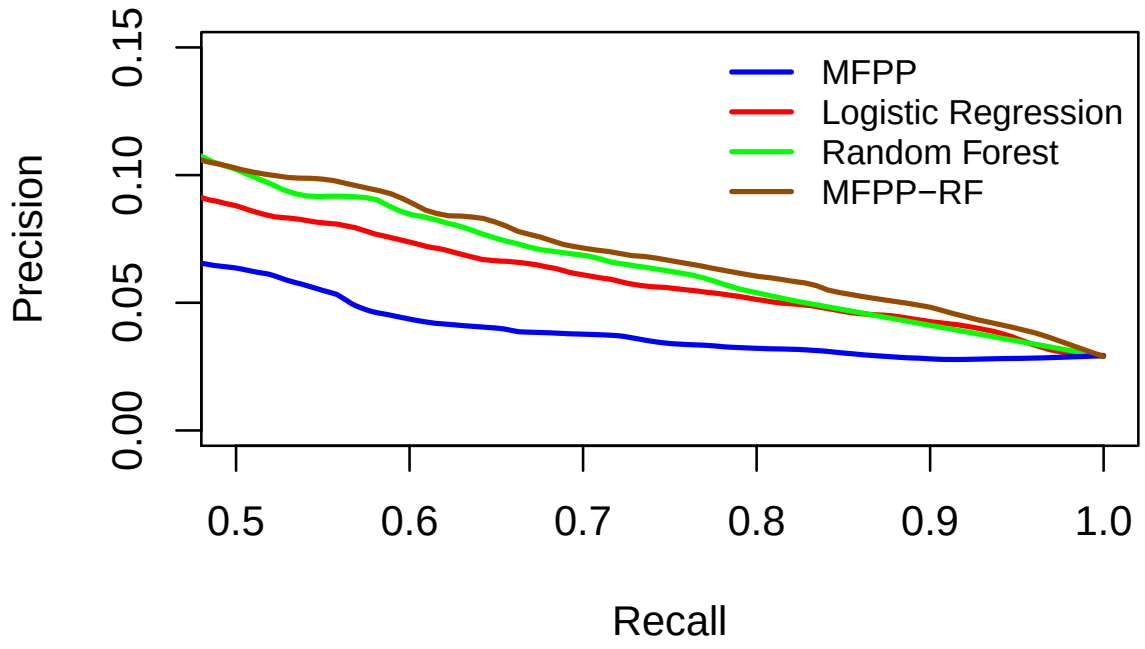

Figure 6.7: Precision-recall curves for supervised forecasting. MFPPs are compared against random forests, logistic regression, and random forests augmented with MFPP intensity features.

2010 given data on subjects prior to 2005. MFPPs and PCIMs perform similarly at this task. The high-recall region is of particular interest in the medical domain because it is more costly to miss a false negative (e.g. undiagnosed heart attack) than a false positive (false alarm). Simply put, clinical practice follows the "better safe than sorry" paradigm, so performance high-recall region is of highest concern. We plot the precision-recall curves between recalls of 0.5 and 1.0 for this reason. The absolute precision for all methods remains low and might exhibit the challenging nature of *ex ante* forecasting. Alternatively, the low precision results could be a result of potential incompatibility of the exponential waiting time assumption and medical event data. Since forecasting can be considered a type of extrapolative prediction, a violation of the model assumptions could lead to suboptimal predictions. Despite these limitations, compared to the baseline precision of $428/15{,}446 = 0.028$, the trained methods do provide utility in forecasting future MI events nonetheless.

Supervised Forecasting Results

Figure 6.7 provides the precision-recall curve for the supervised forecasting experiment predicting at least one myocardial infarction event between 2005 and 2010. As we see, MFPP underperforms compared to all supervised learning methods. However, the MFPP predicted intensities features boosts the MFPP-RF performance compared to the other classifiers. This suggests that while MFPP is a valuable model but may not be optimized for classification.

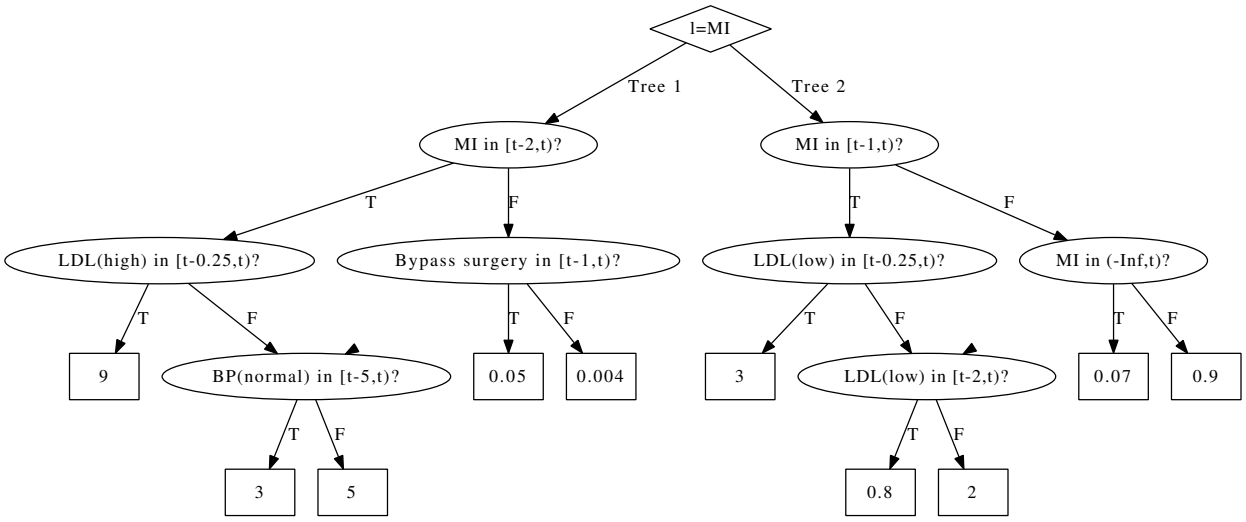

Figure 6.8: First two trees in the MFPP forest. The model shows the rate predictions for myocardial infarction (MI) based on cholesterol (LDL), blood pressure (BP), previous MI, and bypass surgery. Time is in years; for example, [t-1,t) means "within the last year", and (-Inf, t) means "ever before".

MFPPs also provide insight into the temporal progression of events. Figure 6.8 shows the first two trees of the forest learned for the rate of myocardial infarction. We observe the effects on increased risk: history of heart attack, elevated LDL cholesterol levels, abnormal blood pressure, and history of bypass surgery. While the whole forest is not shown (see `http://cs.wisc.edu/~jcweiss/ecml2013/`), the first two trees provide the main effects on the rate. As you progress through the forest, the range over intensity factors narrows towards 1. The tapering effect of relative tree "importance" is a consequence of experimental decision to learn the forest sequentially, and it provides for nice interpretation: the first few trees identify the main effects, and subsequent trees make fine adjustments for the contribution of additional risk factors.

As Figure 6.8 shows, the dominating factor of the rate is whether a recent myocardial infarction event was observed. In part, this may be due to an increased risk of recurrent disease, but also because some EHR events are "treated for" events, meaning that the diagnosis is documented because care is provided. Care for incident heart attacks occurs over the following weeks, and so-called myocardial infarction events may recur over that time frame.

Despite the recurrence effect, the MFPP model provides an interpretable representation of risk factors and their interactions with other events. For example, Tree 1 shows that elevated cholesterol levels increase the rate of heart attack recurrence while normotensive blood pressure measurements decrease it. The findings corroborate established risk factors and their trends.

6.4 Summary

In this chapter we introduced an efficient multiplicative forest learning algorithm to the point process community. We developed this algorithm by combining elements of two continuous-time models taking advantage of their similar likelihood forms. We contrasted the differences between the two models and observed that the multiplicative forest extension of the CTBN framework would integrate cleanly into the PCIM framework. We showed that unlike CTBNs, MFPP forests can be learned independently because of the

PCIM likelihood decomposition and intensity dependence on event history. We applied this model to two data sets: a synthetic model, where we showed significant improvements over the original PCIM model, and a cohort study, where we observed that MFPP-RFs outperformed standard machine learning algorithms at predicting future onset of myocardial infarctions. We provide multiplicative-forest point process code at `http://cs.wisc.edu/~jcweiss/ecml2013/`.

While our work has shown improved performance in two different comparisons, it would also be worthwhile to consider extensions of this framework to marked point processes. Marked point processes are ones where events contain additional information. The learning framework could leverage the information about the events to make better predictions. For example, this could mean the difference between reporting that a lab test was ordered and knowing the value of the lab test. The drawback of immediate extension to marked point processes is that the learning algorithm needs to be paired with a generative model of events in order to conduct accurate forecasting. Without the generative ability, sampled events would lack the information required for continued sampling. The integration of these methods with continuous-state representations would also help allow modeling of clinical events such as blood pressure to be more precise.

Finally, we would like to be able to scale our methods and apply MFPPs to any disease. Because EHR systems are constantly updated, we can acquire new up-to-date information on both phenotype and risk factors. To fully automate the process in the present framework, we need to develop a way to address the scope of the EHR, selecting and aggregating the pertinent features for each disease of interest and identifying the meaningful time frames of interest.

Next we turn to the task of risk attribution. Inspection of learned models, *e.g.*, Figure 6.8, allows us to identify potential risk factors for disease. The next chapter investigates how to quantify the attributable risk specific to individual patients and provides a comparison with the existing clinical paradigm, which applies average results to each study participant.

7 Individualized Risk Attribution from Electronic Health Records

Overview

This chapter focuses on the task of risk attribution from EHR data. Given a disease and a risk factor of interest, *e.g.*, MI and statin use, we seek to quantify how much risk can be attached to the possession of the risk factor. Clinical study paradigms seek to model the average treatment effect (ATE), but tend to apply this population-level effect to future individuals. We argue for the use of the individualized treatment effect (ITE), which has better applicability to new patients, but is harder to reliably estimate. We compare ATE-estimation using randomized and observational analysis methods against ITE-estimation using conditional probability modeling and describe how the ITE theoretically generalizes to new population distributions whereas the ATE may not. On a synthetic data set of statin use and MI, we show that, without access to ground truth, the ITE outperforms the ATE using randomization methodology from Vickers et al. (2007), and, given access to ground truth, improves ITE recovery. We suggest that the conditional probability model should be learned with a consistent, non-parametric algorithm from unweighted examples and show experiments in favor of our argument. The work in this chapter is in preparation for submission.

7.1 Introduction

Randomized controlled trials (RCTs) are the gold standard for determining the risk of a disease attributable to an exposure or treatment. They isolate the effect of a specific treatment on a population by randomization, so that systematic differences in population outcomes can be attributed to the treatment. The primary outcome of an RCT is the average treatment effect (ATE), *i.e.*, the average difference between treatment arms in the probability of the outcome. Because of randomization, the ATE is indicative of effect of treatment even in the presence of other risk factors. The reliability of an RCT conclusion has led to the development of randomization mimics from non-randomized data. These methods manipulate the treatment-outcome frequency estimates of ATE to account for the possibility that the treatment is associated with one (or many) factors causing the outcome but is not the cause itself.

However, when treatments are recommended to future patients, the ATE is not the primary statistic of interest. We do not expect the same treatment effect in every person, and the diversity of effects goes beyond a population's nonuniform prior risk. The belief that treatment effects are individual suggests that we model the individualized treatment effect (ITE), which is the effect of administering the treatment to a person specified in data by a set of recorded features.

Access to the ITE in addition to the ATE is useful in many applications. As discussed in Rothwell (1995), suppose we are considering outcomes of carotid endarterectomy, where our treatment options for carotid stenosis are surgical intervention or watchful waiting. For severe cases of stenosis, the surgery is almost always preferable, while for mild cases, waiting is preferred because of the risks of surgical intervention. Treatment decisions should be individualized because the risk-benefit trade-off will differ according to patient characteristics. Another example is the treatment of borderline-elevated blood pressure, where polypharmacy can become a problem in many individuals with risk factors for type II diabetes and cardiovascular disease

(Kent & Hayward, 2007).

Additionally, recommendations and approvals of drugs change over time. For example, hormone replacement therapy treatment effect findings in RCTs and observational studies were of opposite sign, and advocacy of their use was rescinded when the RCT findings were released (Manson et al., 2013). Similarly, many drugs are taken off the market due to excess harm from adverse drug effects. However, many of these drugs are more effective than alternatives for select populations. ITE modeling can help determine which patients are likely to receive benefit from such drugs and potentially bring drugs back to market safely.

In this chapter, we show the ability to recover the true ITE and the value of the ITE over ATE in synthetic data where we know ground truth. Recovery of the true ITE is theoretically possible provided sufficient data because of algorithmic consistency. We also emphasize another problem of the ATE: its calculation is inherently dependent on the underlying population distribution, when what is desired is a prediction for any new patient independent of the study population. We argue that a non-parametric learning algorithm will recover the conditional probability distribution and do so independently of the population distribution with sufficient data. On synthetic data we show the generalizability of the conditional probability model to alternative population distributions of increasing KL-divergences. We also show that the use of unweighted examples, instead of propensity-score matched examples or stable inverse probability of treatment weighted examples, produces a conditional probability model with a lower MSE for the ITE.

7.2 Background

Randomized controlled trials are the gold standard for estimating the average treatment effect (ATE). The technique randomizes confounders, measured and unmeasured, so that factors important to the disease process are approximately balanced among treatment groups. If measured baseline factors are imbalanced empirically after randomization, propensity scoring can be used obtain covariate balance.

Because RCTs are impractical or infeasible for many exposure-outcome pairs, observational studies were developed to estimate attributable risk. These include studies that use known-confounder modeling, propensity scoring, inverse probability of treatment weighting, and doubly robust estimators (Prentice, 1976; Austin, 2011; Rosenbaum & Rubin, 1983; Robins et al., 2000; Bang & Robins, 2005). The two main ideas in these methods are to (1) adjust for confounders by modeling them, and (2) change the population distribution so that the treatment is independent of confounders given the outcome. One key assumption in all of these models is that there are no unobserved confounders, which is difficult to determine in practice.

Also, in most of these approaches (and their variants), a model is assumed for the conditional probability distribution (CPD) of the outcome given the exposure and covariate. In these cases, the counterfactual outcomes, which are never observed, are assumed to follow the model CPD.

Unlike in ATE estimation, achieving sufficient counts to estimate the counterfactual ITE outcome is infeasible for any moderate-sized feature vector because the number of possible feature states is exponentially large. Therefore, a modeling approach to estimate the counterfactual outcome becomes necessary. These can be the same CPD models used in pseudo-randomized ATE estimation, *e.g.* logistic regression, but in Section 7.3 we will discuss two reasons to adopt other machine learning models: non-uniform treatment recommendations and non-parametric consistency.

Related Work

The call for adoption of the ITE is not new, and the limitations of applying population-average effects on individuals has been noted, *e.g.*, in Kent & Hayward (2007) and Rothwell (1995). The ATE or relative risk is stated as the primary outcome, usually followed by a secondary analysis of the heterogeneity of treatment effect. As mentioned in Hayward et al. (2006), performing subgroup treatment effects is usually more effective in risk-stratified subgroups derived from multivariate analyses than in subgroups defined by individual covariates, and these methods have been adopted for approximating individualized treatment effects (Dorresteijn et al., 2011). While these methods do provide finer-grained treatment effect estimates, factors beyond the outcome risk may influence the treatment effect and can be utilized when modeling the ITE.

Modeling of the ITE has been implemented in several studies. Qian & Murphy (2011) develop the framework of conditional probability modeling and use the predictions to estimate individualized treatment rules (ITRs). Our work builds on this approach, making statements about the utility of the ITE, the generalizability of the ITE, and the preference for using unweighted observational data for ITE estimation, all with simulations to illustrate these advantages. Our simulations based on synthetic data have access to a ground truth ITE, which we use to assess our ITE estimations.

However, it is possible to assess the benefit of ITE without access to ground truth. Vickers et al. (2007) provides an unbiased method to estimate the advantages of using the ITE recommendation over the ATE recommendation using existing RCT data. They show that by counting outcomes in the subset of patients where ITE- and ATE- treatment recommendations disagree, the expected difference in treatment recommendations is estimated. Our experiments include such analyses to show that the ITE-recommendation can be estimated without access to the counterfactual outcomes. Unfortunately, this method can be severely underpowered in the case that the ITE- and ATE- treatment recommendations are highly similar, and a power calculation analysis to determine recruitment size would be helpful. Alternatively, a new RCT study could be implemented randomizing to ITE- and ATE- treatment arms.

Neither the methods we adopt nor the methods presented in Qian & Murphy (2011) directly optimize the ITE. Instead, they model the conditional probability distribution, and then the differences in probability are estimated using the estimates for the treatment effect using true and counterfactual treatments. Zhao et al. (2012) develops a method to directly optimize for the ITE under a surrogate loss function from RCT data. While this method produces ITE recommendations, we believe a model should also provide treatment effect estimates under each treatment arm, because the treatment effect itself is critical information clinically. Also our methods do not require RCT data and scale easily to multiple treatment arms and factorial treatment designs, which are not considered in Zhao et al. (2012).

7.3 Methods

We describe ITE modeling below. Let the ITE for an outcome $y \in \{0, 1\}$ of a patient with features v given treatment $u \in \{0, 1\}$ be the difference in estimates: $p(y = 1|u = 1, v) - p(y = 1|u = 0, v)$. The key assumption made in these modeling approaches is that both the observed outcomes y_{true} and the counterfactual outcomes y_{cf} come from the CPD model, that is, $p(y_{\text{cf}}|u, v) = p(y_{\text{true}}|u, v) = p(y|u, v)$ for all u and v. The interpretation of the ITE is only causal if the no unmeasured confounders assumption (NUCA) is made; otherwise, it is just a statement about the difference in outcome probability given a new patient described by (u, v).

If we have a correctly specified model and NUCA holds, for any new patient with features v and treatment

u, we have their ITE that guides our treatment choice. This statement is notably population-distribution free and thus can generalize to arbitrary population distributions of (u, v). The ATE does not have this characteristic; its calculation is dependent on the distribution of (u, v) so its application should be limited to populations with similar covariate distributions unless the treatment effect is believed to be uniform.

Recalling that the application of the RCT-recommended treatment suggests that every patient should receive that treatment, a logistic-regression-based model similarly provides a uniform decision. Its decision will be in agreement with the sign of the treatment parameter. However, in many cases, and particularly in those where the treatment effect has small magnitude but high variance, the optimal treatment decision is nonuniform. Thus, we adopt machine learning methods which can estimate the CPD while also providing nonuniform treatment choices. In particular, we use AdaBoost because it has consistency results and is a non-parametric learning algorithm (Freund & Schapire, 1996; Culp et al., 2006). In other words, consistency means AdaBoost will recover the correct CPD given enough examples, and will do so regardless of the train (u, v) distribution provided proper support. Non-parametricity allows it to recover any CPD over (u, v), not just ones in a parametric family.

With the adoption of a non-parametric learning algorithm comes the parametric/non-parametric learning trade-off. Parametric models may require smaller sample sizes to learn effectively but are not consistent if misspecified; non-parametric models may require larger sample sizes to achieve good CPD estimates but have consistency results for arbitrary joint distributions.

Recall that propensity-score matching and inverse probability-of-treatment weighting (IPT-W) are methods to produce pseudo-randomized data for the estimation of the ATE. With ITE as the target statistic, these methods become less desirable. In modeling the CPD, propensity score matching and IPT-W weighting reduce the effective sample size, reducing our numbers for estimation. Thus, under the modeling assumption, and, with the goal of modeling ITE, we argue for unweighted CPD estimation.

Experimental Approach

In this section, we restate the claims and reasoning in support of the individualized risk framework and then provide experimental designs to confirm them, using synthetic data with access to ground truth, or observational or RCT data.

As already noted, there is a strong argument for the calculation of the individualized treatment effect (ITE) over the average treatment effect (ATE) because the the ITE can be used in patient-specific recommendations in lieu of ATE-based, population-average recommendations. The value of the ITE recommendation can be estimated, compared against an alternative–for example, the ATE recommendation–using the subsets of randomized patients where treatment recommendation differs (Vickers et al., 2007). We use existing methods to test the hypothesis of ITE superiority and illustrate the benefits of ITE estimation on synthetic data.

We suggest that, in preference for generalizability of study outcome, the conditional probability distribution $p(y|u, v)$ should be modeled with non-parametric learning algorithms. That is, our goal should be to learn the correct $p(y|u, v)$ irrespective of the distribution $p(u, v)$ because future data distributions $p'(u, v)$ may be different. Non-parametric learning algorithms achieve independence from $p(u, v)$ in the limit of increasing data. We empirically characterize the recovery of the ITE varying the train set data size and compare the performance of parametric and non-parametric learners varying the similarity of train and test set population distributions.

Note the relationship to propensity scoring methods, where examples are weighted or matched by a function of $p(u|v)$. Propensity score weighting and matching schemes reduce the effective sample size,

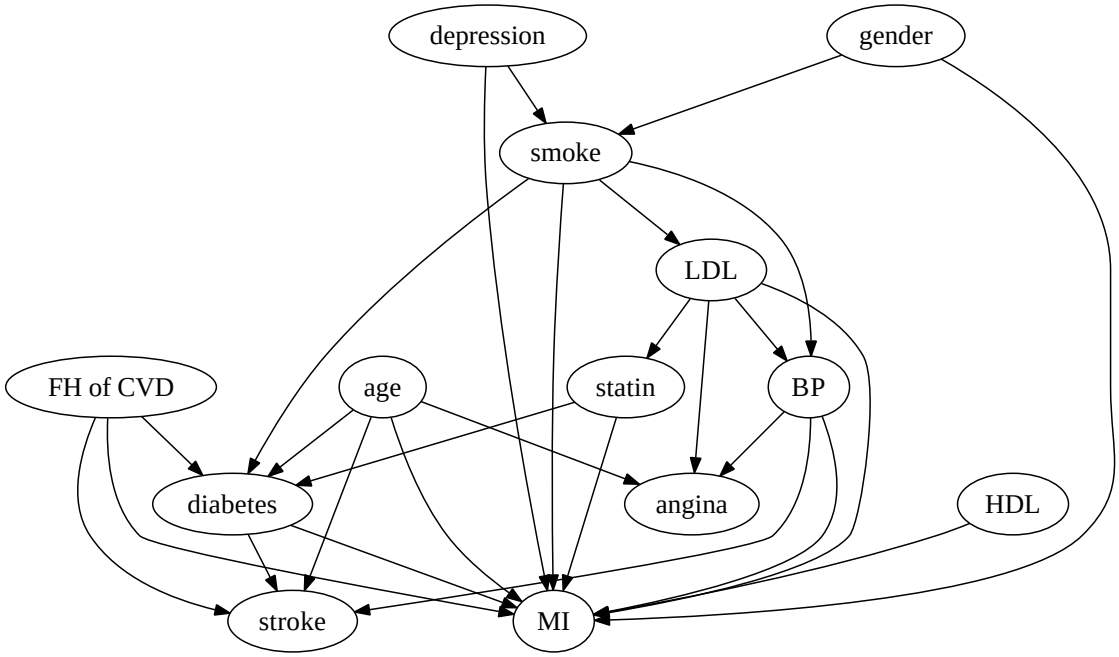

Figure 7.1: Risk attribution model of statin use for MI

mimicking the independence of treatment from observed confounders but not assisting in the recovery of the conditional probability distribution $p(y|u, v)$. We show experimentally that estimating $p(y|u, v)$ directly from the original data distribution outperforms analogous estimators from propensity-score-weighted and stabilized inverse probability-of-treatment weighting methods.

Finally, we discuss applications of the conditional probability distribution modeling approach. Numerous concerns have been voiced about the appropriateness of observational data as a data source for the effect of treatments because confounding can bias the statistical interpretation. With free reign on the covariate definitions in observational studies, we may have access to highly-correlated or even logically-related covariates, such as "ever smoked" and "current smoker." We opt to include such covariates for richness of representation that can lead to better estimates of $p(y|u, v)$, but must adapt our interpretation of "intervention" to specified multivariate changes instead of a (univariate) change of treatment state. We discuss several desired conditions when defining the set of "treatment" states and propose methods to provide interpretable recommendations when the space of "treatment" states is large.

7.4 Experiments

We define two synthetic models of myocardial infarction (MI) with thirteen total variables: age, gender, smoking status, HDL level, LDL level, diabetes status, family history of cardiovascular disease, blood pressure, history of angina, history of stroke, history of depression, statin use, and MI. The network is shown in Figure 7.1. For simplicity, we define each variable to have binary values. The first model–an observational study mimic–uses a Markov Random Field and feature functions corresponding to the edge skeleton of the graph. The second model–an RCT mimic–uses the same graph, and allows us to "intervene" on a variable in the causal network by simply removing the arcs who have the variable as a terminal node and sampling from the new joint distribution. From these models, we can sample synthetic observational and RCT data from the

joint distribution over variables to generate samples, or "patients".

The question we seek to answer is the effect of statin use on heart attack or MI. We test the recommendation from boosted trees against the ATE recommendation on our synthetic, randomized data set, both using the RCT method in Vickers et al. (2007) and comparing against our ground truth knowledge. We use the AdaBoost package in R and default parameter settings to learn the forest (Freund & Schapire, 1996; Culp et al., 2006).

We also compare boosted trees to logistic regression in the observational study setup. We seek to characterize estimation of the ITE under each method by looking at error modes of each model and producing learning curves for the models as a function of training set size. To test for applicability to an arbitrary test population distribution, we alter the distributions on variables with no parents in the causal DAG: age and gender. Finally, we compare ITE and ATE estimation using the unweighted training set with estimation using altered data sets via propensity-score matching and (stabilized) inverse probability-of-treatment weighting.

7.5 Results

Figure 7.2 shows the utility of adopting the ITE recommendation over the ATE recommendation. The upper graph shows that the adoption of the ITE recommendation lowers the probability of MI by 0.0006 on average. Thus, the number-needed-to-treat (NNT) is about 2000, *i.e.*, treating 2000 patients with the ITE-recommended treatment given that the recommendation differs results in one less MI on average. This is a small effect–small due to the fact that there are few patients whose probability of MI would go up with administration of a statin.

The lower graph in Figure 7.2 shows the estimated expected difference in probability of MI between ITE- and ATE- recommended treatments among patients where they disagree on treatment choice. We see that the ITE recommendation lowers the probability of MI in this subset by 0.06, or a NNT of 20.

The learning curves for logistic regression and AdaBoost are shown in Figure 7.3. As we expect, the parametric logistic regression performs well for small train set sizes, but the error cannot approach 0 because the model is misspecified (because the ground truth model is not log-linear in the exposure and covariates). The error of AdaBoost decreases similarly until about 2000 train set examples, where it continues to reduce the MSE towards 0. The approach toward 0 error is in line with the non-parametric consistency results that exist for AdaBoost (Bartlett & Traskin, 2007).

Figure 7.4 shows the error modes for ITE estimations using logistic regression and AdaBoost; the errors are smaller using AdaBoost. The plots show the estimated ITE versus the ground truth ITE for the test set examples in black with the ATE applied to all examples overlaid in red. Having all points on the line $y = x$ is optimal. For logistic regression (top), all ITE estimates will be either above or below 0 because the model assumes that a single coefficient determines the direction of the effect. AdaBoost does not have this restriction and can provide individualized recommendations, though, as is shown, the errors are still non-zero.

The effect of different data-weighting and matching schemes is shown in Figure 7.5. The recovery of the CPD model and thus the ITE requires the fewest examples by leaving the examples unweighted, more using stabilized inverse weighting, and the most using 1:1 propensity score matching. One important consideration is that our data set includes some patients without elevated LDL who take statins, motivated by the suggestion that there could be therapeutic benefit of statins even in borderline hypercholesterolemia. However, in a data set with few normal-LDL statin users, propensity-score matching and particularly stabilized inverse weighting will impair the CPD model, because it will attach large excess weight to few examples.

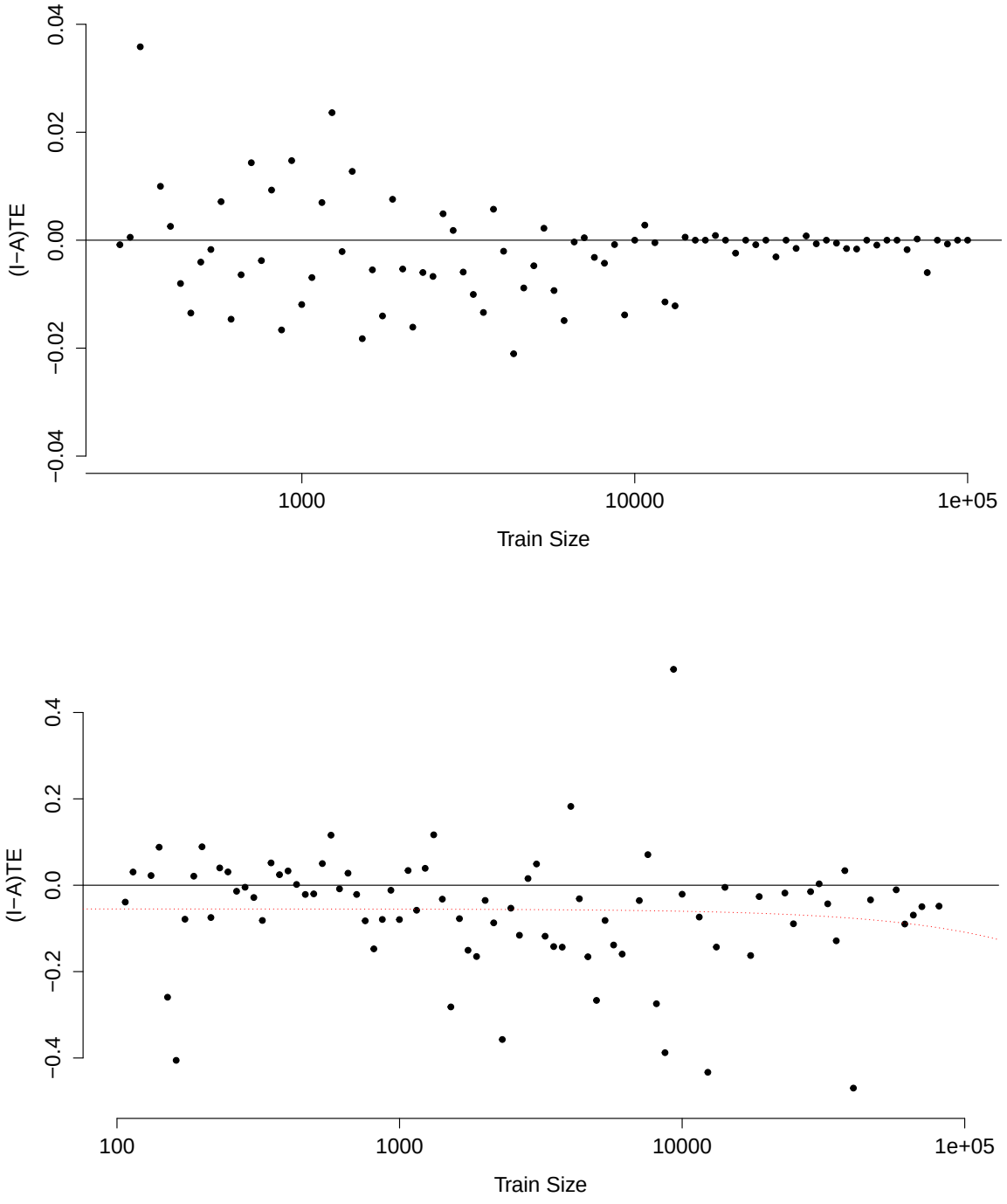

Figure 7.2: Average difference in treatment effect using the ITE recommendation in place of the ATE recommendation as a function of train set size. The estimated difference in the population is shown at top; the estimated difference in the subpopulation where treatment recommendations differ is shown at bottom. The red dotted line indicates the least square fit. The difference in treatment effect is estimated by the Vickers et al. (2007) method with a test set of 50000 examples.

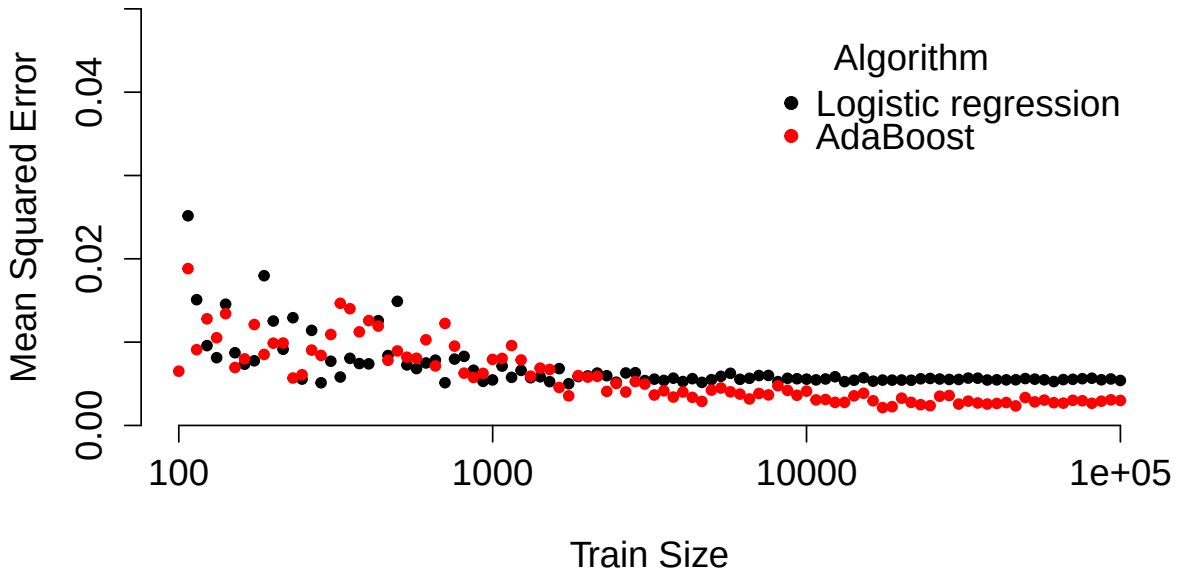

Figure 7.3: Learning curves for logistic regression (black) and AdaBoost (red); test set ITE mean-squared error as a function of training set size.

Shifting the test set distribution by adjusting the prevalence of the young-and-female to old-and-male subgroups had no substantial effect on the difference in AdaBoost and logistic regression MSE of the ITE. This is surprising because the nonparametric AdaBoost would be expected to generalize to alternative distributions better than the parametric logistic regression. It is possible that age and gender do not influence the ITE of subgroups differently, suggesting that looking for heterogeneity of treatment effect in risk-based strata (age and gender are risk factors for MI) may not detect underlying treatment effect differences.

7.6 Discussion

The work presented in this chapter is in preparation and will require empirical justification in several directions.

- We seek to apply our framework to real clinical data: both to RCT and observational data. We intend to use International Stroke Trial (RCT) and Marshfield Clinic Personalized Medicine Research Project (EHR cohort) data (Sandercock et al., 2011; McCarty et al., 2005). In these settings, we will not have access to the ground truth. Nevertheless, we can adopt the approach in Vickers et al. (2007) to compare ITE and ATE outcomes. For evaluation, we must resort to the average outcome over some population, preferably several populations with covariate distributions different from each other and the training population. A characterization of which (u, v) provide reliable ITE recommendations is critical as well. There may be more uncertainty in patients underrepresented in the train distribution, especially for limited train set sizes.

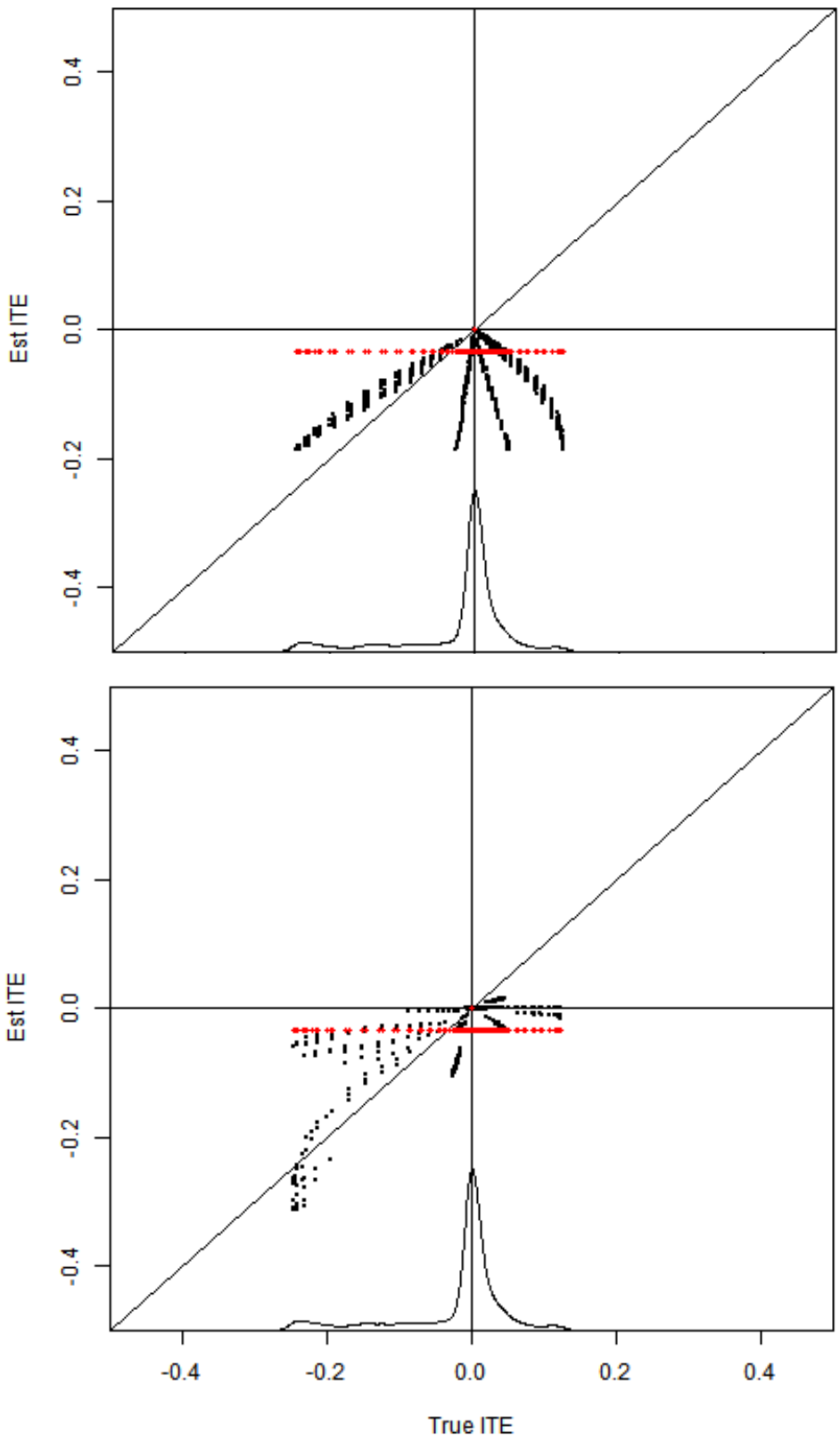

Figure 7.4: Estimated ITE (black) and ATE (red) versus the ground truth ITE for logistic regression (top) and AdaBoost (bottom) for a train set size of 50000. Optimal estimation is given by the line $y = x$. Empirical smoothed density of the ground truth ITE is shown at bottom.

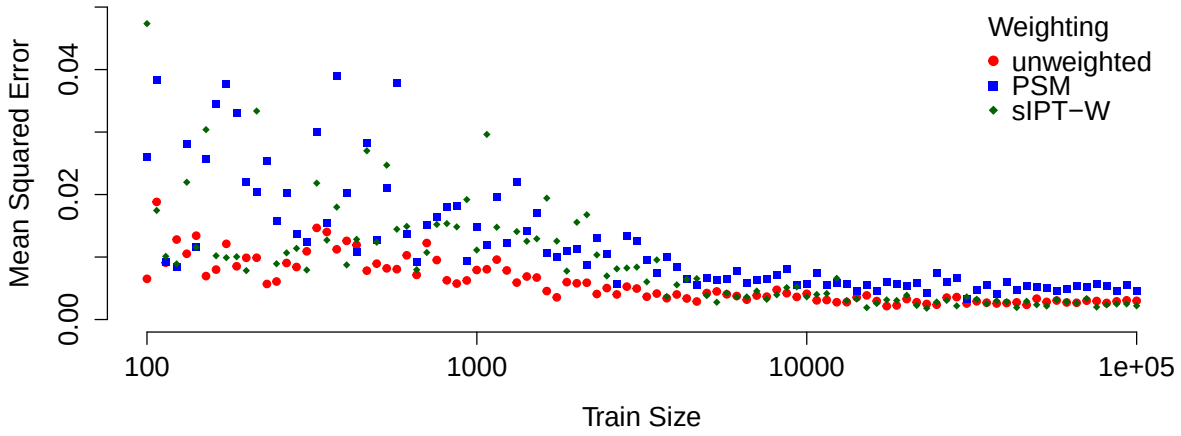

Figure 7.5: Learning curves for AdaBoost using unweighted examples (red circles), propensity-score matched samples with a 1:1 ratio (blue squares), and stabilized inverse weighted examples (green diamonds): test set ITE mean-squared error as a function of training set size.

- We want to characterize what train set size is needed for non-parametric learning algorithms to outperform the parametric algorithms and specifically logistic regression. A characterization of the number of examples needed to move past the mean squared error of the logistic regression outcome for a given task is important as a factor in determining when we should recommend the model ITE outcome or stick with the ATE.
- Using the Vickers et al. (2007) method, we would like to characterize our uncertainty in our point estimates–ITEs for patients–as well as perform power calculations to determine sample sizes needed to detect ITE superiority.

Another crucial issue preventing the automatic use of EHR data for risk attribution is the presence of unobserved confounders, observed confounders, and intermediate variables. For a pure prediction problem, it makes sense to use all features that carry information useful in prediction of the outcome, provided a large enough training population. For risk attribution however, the exclusion of a confounder or the inclusion of an intermediate variable can result in biased estimation of both the ITE and ATE. However, we should not simply accept a regime that excludes intermediate variables, because the inclusion of intermediate variables may enhance our modeling of the conditional probability distribution. At test time, we may simply not have access to the intermediate variables and could instead have to infer their values and produce a Bayesian estimate for the resulting ITE.

Issues with intermediate variables and confounders arise in EHR data also because of the multitude and specificity of variable definitions. For example, suppose we have two logically related features: "history of smoking" and "current smoker". Clearly these variables are intertwined and potentially useful for outcomes, *e.g.*, lung cancer. When we ask what the risk attributable to smoking is, we need to be more specific as to which variable we mean. Suppose we choose "history of smoking". Most clinical analyses would then omit "current smoker" from the analysis, despite its importance as a lung cancer predictor. The support for its

removal is that it is an intermediate variable, or alternatively the study design might look at the effect of "history of smoking" among the subpopulations of current and former smokers.

For confounders, *e.g.*, when the exposure of interest is "alcohol consumption," clinical analyses will typically include one but not both features in the model. The inclusion of one feature allows the model to control for smoking, *i.e.*, that smoking behavior is associated with alcoholism and causes lung cancer. Both features are not included to limit the degrees of freedom in the covariates to explain away the effect of exposure. In either case, study design opts for the removals of features that improve the conditional probability model.

Because of these challenges, we suggest several approaches to explore:

- For interventions, we can define the scope of their feature influence, potentially probabilistically, and potentially including multiple features. For example, a diabetes intervention could be represented in the replacement of feature values for "rosaglitazone", "fasting blood sugar" and "HbA1c". Then we can model the effect of intervention by comparing probability of outcomes under intervention or no intervention while richly modeling the conditional probability distribution.

- A timeline-based analysis where the effect of the intervention on apparent intermediate variables as well as the outcome of interest could be modeled and hence improve outcome prediction.

Finally, though modeling of the ITE is enticing, robustness guarantees and validation of its performance should be established before large-scale clinical deployment. A few sources of validation include replication studies and heterogeneity of treatment effect analyses using ITE model strata.

7.7 Summary

In this chapter, we illustrated the parallels between the standard clinical study framework designed to determine the ATE and the burgeoning clinical study framework designed to determine the ITE. We first argued that the ATE is favorable in its ability to leverage the RCT study design. Then, we highlighted shortcomings of the ATE, first, that the ATE is an average outcome, when in practice we usually care about the ITE for future patients, and second, that the ATE is population-distribution dependent. Then we discussed modeling of the ITE. Notably the logistic regression can only recommend one treatment arm if we exclude non-linear and exposure-covariate interaction terms because the coefficient for exposure is either negative or non-negative. Furthermore, unless correctly-specified, the logistic regression is not a consistent learning algorithm, so we cannot hope to recover the true conditional probability distribution even from large populations. Instead, we adopted another popular framework, boosted trees, and showed that the forest-based ITE outperformed the ATE on a synthetic problem of MI prediction. Finally, we showed that the use of propensity-score matching and inverse-probability-of-treatment weighting impaired the learning of the conditional probability distribution, so we recommend against the use of PSM and IPT-W unless you are interested in an estimate for the ATE in a pseudo-randomized population.

8 Conclusions

In this thesis we presented foundations for statistical timeline analysis (STA): the applications of temporal and relational modeling to answer predictive and risk attribution questions from Electronic Health Records. The need for STA comes from the limitations of existing methods at modeling the nascent framework of the EHR as a data source for clinical modeling. These limitations come from the difficulty in addressing the challenges of EHR data: relational, temporal, noisy, and incomplete not at random. We argued that patient data could be effectively represented as timelines, and we developed methods to address aspects of timeline data.

8.1 Contributions

In Chapter 3 we tackled the difficult problem of predicting primary MI from EHR data at the Marshfield Clinic using a subset of known risk factors. We found that two SRL algorithms outperformed their propositional analogues suggesting the utility of relational learning algorithms for EHR timeline analysis. The RFGB forest learning method performed the best of any algorithm, particularly in the high recall region of the precision-recall curve. Contrary to most clinical studies, important predictive features outside of the "patient-disease" framework are also found, such as implicit physician awareness of disease observed in testing, and relational information tying patients, physician and providers that hints at a patient's medical condition but is not measured in specified covariates.

We also stated several limitations to our RFGB experimentation. We did not use additional relational information available to us, such as hierarchical diagnosis relationships, prescription relationships, and family relationships. Decisions about the data representation would be necessary because the representation affects the RFGB search behavior over relational features. Also, the use of timeline information in RFGB is limited to logical, temporal comparison features, such as "did a hypertensive event within the last year also precede a prescription of a beta blocker"? That the features switch from on to off in an instant seems medically unreasonable, and the approaches modeling the instantaneous rate of events like CTBNs and point processes move the discontinuities into the rates of events instead.

In Chapter 4, we turned to continuous-time Bayesian networks (CTBNs) for timeline modeling. We reviewed the CTBN, an elegant mathematical model for describing a distribution of a set of discrete variables over continuous time. We noted that it compactly represents the exponential-size joint state by modeling the rate of state changes individually for each variable, where the rates are dependent on the state of parent variables, which themselves are specified by a graphical model over the variables. Unfortunately, while a CTBN is a compact representation of the joint state, the size of the model still scales exponentially in the number of parents of a variable. This means, at learning time, that an exponential number of parameters must be estimated.

Our work in Chapter 4 introduced a generalization of CTBNs: partition-based CTBNs. For each variable, the rate is determined by one part of the partition, and the number of parts can be specified arbitrarily. Our work showed that trees and forests could be used to represent the partitions, and that the number of parameters for such a model would be linear (not exponential) in the number of splits in the forest. Then, we showed that the forests could be learned efficiently using a maximum likelihood approach and a multiplicative rate assumption. By addressing the limitation of scalability in parameter size, we showed

experimentally that CTBN models on the order of hundreds of variables could be learned effectively.

Chapter 5 addressed the problem of CTBN inference. Despite the efficient learning algorithm provided in Chapter 4, when the input data is missing the state of any variable for any interval duration, inference must be performed to probabilistically complete the interval. EHR data serves as point event data so it tends to missingness in the extreme because at any given time t, the number of events observed is zero with near certainty. Thus CTBN inference is needed to "fill in" the timeline. A variety of CTBN inference algorithms have been proposed, but none scale to EHR-sized data. Chapter 5 followed the sampling and particle filtering approach presented in Fan & Shelton (2008). In particular, it introduced a general method of improvement to sequential importance sampling and applied the solution to CTBN inference. The improvement is that, given a target distribution f and surrogate distribution g from which we can sample, we selectively reject a portion of the sample steps so that the weight distribution of samples from g has lower variance, which means that we can approximate f equally well with fewer samples. We showed that the decision of sample rejection can be learnt using weighted samples and a binary classifier that encodes the model and the evidence. Experimentally, we showed that learning a logistic regression model for sample rejection improved CTBN inference by an order of magnitude, assuming a willingness to pay the up front cost to train the classifier. Extensions of this line of reasoning to achieve greater speed-ups should be investigated because more than an order of magnitude scale-up will be necessary to apply CTBNs and CTBN inference on problems with thousands or millions of variables seen in EHR data.

Given existing challenges in CTBN inference, in Chapter 6 we investigated an alternative model for timelines: point processes. We sought to translate our learning contributions in CTBNs to the point process framework and showed that it was possible if we used the piecewise-constant intensity model (PCIM) framework. We connected CTBN and PCIM frameworks by noting their similarity in likelihood formulations and extended PCIMs (Gunawardana et al., 2011) to forest models. We showed experimentally that learning forests improved the model likelihood and that features derived from the forests improved prediction in a variety of forecasting tasks for myocardial infarction. A few key differences between CTBN and point processes were identified, such as the ability to learn models for each point process variable separately, due to the lack of inference required and to the likelihood decomposition. Thus, in practice, one could learn a rate dependence for individual outcomes of interest, instead of for every variable in the model. At the same time, point processes are restricted in their ability to model clinical processes because of their strong assumptions that the observation of an event equates to the occurrence of said event, and that the lack of observation of an events means the event did not occur (*i.e.*, the closed-world assumption).

Chapter 7 transitioned focus to the task of risk attribution. While mature methods are widely adopted for determining the average treatment effect (ATE) using randomized controlled trials, cohort studies and case-control analyses, methods focusing on individualized treatment effect (ITE) are just now being utilized. Estimation of the ITE requires a modeling approach, which lends itself perfectly as a machine learning task. Furthermore, because of the impracticality of performing a trial for exposure-outcome pair of interest, modeling using EHR data becomes especially attractive. With these motivations, we investigated the use of EHR data to determine a patient's personalized risk attributable to an exposure or treatment of choice. To motivate use of the ITE, we discussed the severe limitation of the applicability of the ATE. We suggested that the ITE, unlike the ATE, could generalize to arbitrary population distributions. We showed that a choice of binary classifiers different than logistic regression or a modeling of exposure-interaction terms would be necessary for modeling the ITE in cases where there exists heterogeneity of treatment recommendation. On a synthetic data set of statin treatment for MI prevention, we found that the personalized ITE recommendation outperformed the uniform ATE recommendation. We also described learning curves and error modes for

logistic regression and boosted trees, and showed that boosted trees outperformed logistic regression in ITE recovery. Finally we showed that learning from unweighted data instead of from propensity-score matched or inverse-probability-of-treatment weighted data results improves our ability to model the conditional probability distribution.

8.2 Future Work

Discussions of future work specific to each chapter are contained in the chapters themselves; here we discuss future directions for statistical timeline analysis as a whole.

Chapters 4, 5, and 6 hold the core temporal components of our work. We showed that the forest learning algorithms can effectively recover temporal dependencies in data, which can lead to improved prediction. However, as illustrated in Chapter 6 learning temporal dependency models does not optimize for predictions of future outcomes. If the network we learn is not causal, or there are unobserved confounders or effect modifiers, we should not hope that forward sampling from our models should make accurate predictions beyond the near future. Future work in learning temporal, causal models could be difficult particularly because of the need to address unobserved confounders and effect modifiers. However, a characterization of the causal effect of variables in the system could determine a bound on the certainty of the forecast as a function of time into the future.

The simpler approach is to make the assumption that the future will behave like the past, so we can perform supervised learning on training examples from the past and apply them to examples in the future. While this is straightforward and has immediate application, fields like medicine are always undergoing rapid shifts in the practice of care and the definitions of disease, so inferring about the future using the past will require constant relearning and will never get away from some small temporal bias. Regardless of the methodology chosen, our temporal analysis does show improvement in performance over existing models, suggesting that capturing the temporal dependencies is important both for forecasting and for characterization of underlying processes.

This thesis presented a wide range of algorithms, which were developed to address key shortcomings of existing methods when applied to the analysis of EHR data. Future work should investigate the combination or hybridization of these models. Relational temporal models such as marked point processes or relational CTBNs is worth exploring to better capture the relational nature of EHR data while maintaining an appropriate continuous-time representation. A straightforward extension to learning comes to mind, though inference in both models could be challenging. CTBN inference will undoubtedly have scaling challenges when applied to relational data, though the use of, *e.g.*, lifted particle filters Nitti et al. (2013) could mitigate it. Sampling of marked point processes will require a relational generator in order to forecast, so its use to undercover interesting relational dependencies may provide more immediate utility.

Hybrid forest CTBNs and point processes is a natural extension of our work. The limitation of forest point processes is that inference must be based on forward sampling, *i.e.*, sequential importance sampling, filtering and smoothing. However, strong potential for scaling up such inference methods was presented in Chapter 5.

The combination of ITE estimation presented in Chapter 7 with temporal modeling is important and one of the most challenging combination of themes. The ITE by definition is a difference in the individualized probability of an outcome, and in many cases, the outcome value is defined by the presence or absence of a disease over a time duration. A temporal ITE might then take a functional form, *i.e.*, the instantaneous difference in rate of the outcome occurring under the treatment and non-treatment arms. A temporal ITE defined this way is similar to tracking the difference in intensities in a CTBN or point process over time.

Establishing theoretical guarantees about a temporal ITE and how it could affect treatment choice would be important next steps. The temporal ITE also bears some resemblance to the Cox proportional-hazards model (Cox et al., 1972), which calculates the average attributable rate difference over time series data using a semi-parametric model, though again, the modeling of individualized effects has distinct advantages over population-average effects as described in Chapter 7.

Finally, returning to the medical utility of statistical timeline analysis for EHR data, the methods described here could be the precursors to an automated system that provides clinical assessments of patients based on data that is already collected for medical record-keeping and billing services. Additional pertinent information can be introduced when clinical suspicion is high to update the clinical assessments, but the ability of the EHR to provide clinical, statistical, and individualized guidance can improve patient care. A system deployed now would not be perfect by any means, but observing its performance with the existing methodology would highlight the improvements necessary to produce changes in clinical practice with statistically minded improvements in outcome.

List of References

Ahmadi, B., Kersting, K., and Natarajan, S. Lifted online training of relational models with stochastic gradient methods. *Machine Learning and Knowledge Discovery in Databases*, pp. 585–600, 2012.

Anderson, G. and Pfahringer, B. Relational Random Forests Based on Random Relational Rules. In *Proceedings of the Interational Joint Conferences in Artificial Intelligence (IJCAI)*, 2009.

Andrieu, Christophe, Doucet, Arnaud, and Holenstein, Roman. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Antonopoulos, S. Third Report of the National Cholesterol Education Program (NCEP) Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults (Adult Treatment Panel III) Final Report. *Circulation*, 106(3143):3421, 2002.

Austin, Peter C. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46(3):399–424, 2011.

Bang, Heejung and Robins, James M. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

Bartlett, Peter L and Traskin, Mikhail. Adaboost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.

Blockeel, H. and Raedt, L. De. Top-Down Induction of First-Order Logical Decision Trees. *Artificial Intelligence*, 101:285–297, 1998.

Bog-Hansen, E., Larsson, C. A., Gullberg, B., Melander, A., Bostrom, K., Rastam, L., and Lindblad, U. Predictors of Acute Myocardial Infarction Mortality in Hypertensive Patients Treated in Primary Care. *Scandinavian Journal of Primary Health Care*, 25(4):237–243, 2007.

Breiman, L. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Brown, E. N., Kass, R. E., and Mitra, P. P. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.

Cavallo, R. and Pittarelli, M. The theory of probabilistic databases. In *Proceedings of the 13th International Conference on Very Large Data Bases*, pp. 71–81, 1987.

Cheng, Jian and Druzdzel, Marek J. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13(1):155–188, 2000.

Chopin, Nicolas, Jacob, P, Papaspiliopoulos, Omiros, et al. Smc2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. *JR Stat. Soc. B (2012, to appear)*, 2011.

Cohn, I., El-Hay, T., Friedman, N., and Kupferman, R. Mean field variational approximation for continuous-time Bayesian networks. In *Uncertainty in Artificial Intelligence*, 2009.

Cornebise, J., Moulines, E., and Olsson, J. Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480, 2008.

Cox, David R et al. Regression models and life tables. *JR stat soc B*, 34(2):187–220, 1972.

Craven, M. and Shavlik, J. Extracting Tree-Structured Representations of Trained Networks. In *Proceedings of the Neural Information Processing Systems (NIPS) Conference*, pp. 24–30, 1996.

Culp, Mark, Johnson, Kjell, and Michailidis, George. ada: An r package for stochastic boosting. *Journal of Statistical Software*, 17(2):9, 2006.

Dean, T. and Kanazawa, K. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, 1989.

Dietterich, T.G., Ashenfelter, A., and Bulatov, Y. Training Conditional Random Fields via Gradient Tree Boosting. In *Proceeding of the International Conference on Machine Learning (ICML)*, 2004.

Diggle, P.J. and Rowlingson, B.S. A conditional approach to point process modelling of elevated risk. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pp. 433–440, 1994.

Diverse Populations Collaborative Group. Prediction of Mortality from Coronary Heart Disease among Diverse Populations: Is There a Common Predictive Function? *Heart*, 88:222–228, 2002.

Dorresteijn, Johannes AN, Visseren, Frank LJ, Ridker, Paul M, Wassink, Annemarie MJ, Paynter, Nina P, Steyerberg, Ewout W, van der Graaf, Yolanda, and Cook, Nancy R. Estimating treatment effects for individual patients based on the results of randomised clinical trials. *BMJ*, 343, 2011.

Doucet, A., Godsill, S., and Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Engle, R.F. The econometrics of ultra-high-frequency data. *Econometrica*, 68(1):1–22, 2000.

Fan, Yu and Shelton, Christian R. Sampling for approximate inference in continuous time Bayesian networks. In *Tenth International Symposium on Artificial Intelligence and Mathematics*, 2008.

Fan, Yu, Xu, Jing, and Shelton, Christian R. Importance sampling for continuous time Bayesian networks. *The Journal of Machine Learning Research*, 11:2115–2140, 2010.

Freund, Y. and Schapire, R. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, 1995.

Freund, Y. and Schapire, R. Experiments with a new boosting algorithm. In *ICML*, 1996.

Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pp. 1189–1232, 2001.

Getoor, L. and Taskar, B. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, 2007.

Greenland, P., Alpert, J. S., Beller, G. A., Benjamin, E. J., Budoff, M. J., Fayad, Z. A., Foster, E., Hlatky, M., Hodgson, J. M. B., and Kushner, F. G. 2010 ACCF/AHA Guideline for Assessment of Cardiovascular Risk in Asymptomatic Adults. *Journal of the American College of Cardiology*, pp. j. jacc. 2010.09. 001v1, 2010.

Gunawardana, A., Meek, C., and Xu, P. A model for temporal dependencies in event streams. Advances in Neural Information Processing Systems, 2011.

Gutmann, B. and Kersting, K. TildeCRF: Conditional Random Fields for Logical Sequences. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2006.

Hall, Mark, Frank, Eibe, Holmes, Geoffrey, Pfahringer, Bernhard, Reutemann, Peter, and Witten, Ian H. The WEKA Data Mining Software: an Update. *Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter*, 11(1):10–18, 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL `http://dx.doi.org/10.1145/1656274.1656278`.

Hayward, Rodney A, Kent, David M, Vijan, Sandeep, and Hofer, Timothy P. Multivariable risk prediction can greatly enhance the statistical power of clinical trial subgroup analysis. *BMC Medical Research Methodology*, 6(1):18, 2006.

Heckerman, D. Causal independence for knowledge acquisition and inference. In *UAI*, pp. 122–127, 1993.

Heritage Provider Network, Inc. Heritage Health Prize. `http://www.heritagehealthprize.com/c/hhp`, June 2011.

Jensen, D. and Neville, J. Linkage and autocorrelation cause feature selection bias in relational learning. In *Machine Learning-International Workshop then Conference-*, pp. 259–266. Citeseer, 2002.

Jha, Ashish K. Meaningful use of electronic health records: the road ahead. *JAMA*, 304(15):1709–1710, 2010.

Kannel, W.B. Blood pressure as a cardiovascular risk factor. *JAMA*, 275(20):1571, 1996.

Kay, Misha, Santos, Jonathan, and Takane, Marina. mhealth: New horizons for health through mobile technologies. *World Health Organization*, 2011.

Kent, David M and Hayward, Rodney A. Limitations of applying summary results of clinical trials to individual patients: the need for risk stratification. *JAMA*, 298(10):1209–1212, 2007.

Kersting, K. and Driessens, K. Non–Parametric Policy Gradients: A Unified Treatment of Propositional and Relational Domains. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

Kong, Augustine, Liu, Jun S, and Wong, Wing Hung. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

Liu, Jun S, Chen, Rong, and Wong, Wing Hung. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):1022–1031, 1998.

Manson, J. A. E., Tosteson, H., Ridker, P. M., Satterfield, S., Hebert, P., O'Connor, G. T., Buring, J. E., and Hennekens, C. H. The Primary Prevention of Myocardial Infarction. *New England Journal of Medicine*, 326 (21):1406–1416, 1992.

Manson, JoAnn E, Chlebowski, Rowan T, Stefanick, Marcia L, Aragaki, Aaron K, Rossouw, Jacques E, Prentice, Ross L, Anderson, Garnet, Howard, Barbara V, Thomson, Cynthia A, LaCroix, Andrea Z, et al. Menopausal hormone therapy and health outcomes during the intervention and extended poststopping phases of the women's health initiative randomized trials. *JAMA*, 310(13):1353–1368, 2013.

McCarty, C. A., Wilke, R. A., Giampietro, P. F., Wesbrook, S. D., and Caldwell, M. D. Marshfield Clinic Personalized Medicine Research Project (PMRP): design, methods and recruitment for a large population-based biobank. *Personalized Medicine*, 2(1):49–79, 2005.

McCarty, C. A., Peissig, P., Caldwell, M. D., and Wilke, R. A. The Marshfield Clinic Personalized Medicine Research Project: 2008 scientific update and lessons learned in the first 6 years. *Personalized Medicine*, 5(5): 529–542, 2008.

Montemerlo, Michael, Thrun, Sebastian, Koller, Daphne, and Wegbreit, Ben. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, volume 18, pp. 1151–1156. Lawrence Erlbaum Associates LTD, 2003.

Natarajan, S., Khot, T., Kersting, K., Guttmann, B., and Shavlik, J. Boosting Relational Dependency networks. In *Inductive Logic Programming*, 2010.

Natarajan, S., Joshi, S., Tadepalli, P., Kristian, K., and Shavlik, J. Imitation Learning in Relational Domains: A Functional-Gradient Boosting Approach. In *Proceedings of the Interational Joint Conferences on Artificial Intelligence (IJCAI)*, 2011a.

Natarajan, S., Khot, T., Kersting, K., Guttmann, B., and Shavlik, J. Gradient-based Boosting for Statistical Relational Learning: The Relational Dependency Network Case. *Machine Learning*, 2011b.

Neville, J., Jensen, D., Friedland, L., and Hay, M. Learning Relational Probability Trees. In *Proceedings of the Knowledge Discovery and Data Mining (KDD) Conference*, 2003.

Nitti, Davide, De Laet, Tinne, and De Raedt, Luc. A particle filter for hybrid relational domains. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 2764–2771. IEEE, 2013.

Nodelman, U., Shelton, C. R., and Koller, D. Learning continuous time Bayesian networks. In *Uncertainty in Artificial Intelligence*, 2003.

Nodelman, U.D. *Continuous time Bayesian networks*. PhD thesis, Stanford University, 2007.

Nodelman, Uri, Koller, Daphne, and Shelton, Christian R. Expectation propagation for continuous time Bayesian networks. In *Uncertainty in Artificial Intelligence*, 2005.

Prentice, Ross. Use of the logistic model in retrospective studies. *Biometrics*, pp. 599–606, 1976.

Qian, Min and Murphy, Susan A. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.

Raedt, L.D. Probabilistic logic learning. *Logical and Relational Learning*, pp. 223–288, 2008.

Rajaram, S., Graepel, T., and Herbrich, R. Poisson-networks: A model for structured point processes. In *AI and Statistics*, 2005.

Rao, Vinayak and Teh, Yee Whye. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Uncertainty in Artificial Intelligence*, 2011.

Ripley, B.D. The second-order analysis of stationary point processes. *Journal of Applied Probability*, pp. 255–266, 1976.

Robins, James M, Hernan, Miguel Angel, and Brumback, Babette. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

Rosenbaum, Paul R and Rubin, Donald B. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Rothwell, Peter M. Can overall results of clinical trials be applied to all patients? *The Lancet*, 345(8965): 1616–1619, 1995.

Sandercock, Peter AG, Niewada, Maciej, and Członkowska, Anna. The International Stroke Trial database. *Trials*, 12(1):1–7, 2011.

Saria, S., Nodelman, U., and Koller, D. Reasoning at the right time granularity. In *UAI*, 2007.

Shelton, C.R., Fan, Y., Lam, W., Lee, J., and Xu, J. Continuous time Bayesian network reasoning and learning engine. *JMLR*, 11:1137–1140, 2010.

Simma, Aleksandr. *Modeling Events in Time Using Cascades Of Poisson Processes*. PhD thesis, EECS Department, University of California, Berkeley, Jul 2010.

Strobl, C., Malley, J., and Tutz, G. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, 14(4):323, 2009.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the Neural Information Processing Systems (NIPS) Conference*, 2000.

Tzoulaki, I., Liberopoulos, G., and Ioannidis, J. Assessment of claims of improved prediction beyond the Framingham risk score. *JAMA*, 302(21):2345, 2009.

Vickers, Andrew J, Kattan, Michael W, and Sargent, Daniel J. Method for evaluating prediction models that apply the results of randomized trials to individual patients. *Trials*, 8(1):14, 2007.

Weiss, Jeremy C and Page, David. Forest-based point process for event prediction from electronic health records. In *Machine Learning and Knowledge Discovery in Databases*, pp. 547–562. Springer, 2013.

Weiss, Jeremy C., Natarajan, S., Peissig, P.L., McCarty, C.A., and Page, D. Statistical relational learning to predict primary myocardial infarction from electronic health records. In *IAAI*, 2012a.

Weiss, Jeremy C., Natarajan, S., Peissig, P.L., McCarty, C.A., and Page, D. Machine learning for personalized medicine: Predicting primary myocardial infarction from electronic health records. *AI Magazine*, 33(4):33, 2012b.

Weiss, Jeremy C., Natarajan, Sriraam, and Page, David. Multiplicative forests for continuous-time processes. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2012c.

Wilson, P.W.F., D'Agostino, R.B., Levy, D., Belanger, A.M., Silbershatz, H., and Kannel, W.B. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

Wolfel, Matthias and Faubel, Friedrich. Considering uncertainty by particle filter enhanced speech features in large vocabulary continuous speech recognition. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pp. IV–1049. IEEE, 2007.

Xu, Jing and Shelton, Christian R. Intrusion detection using continuous time Bayesian networks. *Journal of Artificial Intelligence Research*, 39(1):745–774, 2010.

Yuan, Changhe and Druzdzel, Marek J. An importance sampling algorithm based on evidence pre-propagation. In *Uncertainty in Artificial Intelligence*, pp. 624–631. Morgan Kaufmann Publishers Inc., 2003.

Yuan, Changhe and Druzdzel, Marek J. Importance sampling for general hybrid Bayesian networks. In *Artificial Intelligence and Statistics*, 2007a.

Yuan, Changhe and Druzdzel, Marek J. Improving importance sampling by adaptive split-rejection control in Bayesian networks. In *Advances in Artificial Intelligence*, pp. 332–343. Springer, 2007b.

Zhao, Yingqi, Zeng, Donglin, Rush, A John, and Kosorok, Michael R. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.