Gaussian Process-Based Models for Clinical Time Series in Healthcare

by

Joseph Futoma

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Katherine Heller, Supervisor

_____
David Dunson

_____
Jerome Reiter

_____
Joseph Lucas

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2018

Abstract

Gaussian Process-Based Models for Clinical Time Series in Healthcare

by

Joseph Futoma

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Katherine Heller, Supervisor

_____
David Dunson

_____
Jerome Reiter

_____
Joseph Lucas

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2018

Abstract

Clinical prediction models offer the ability to help physicians make better data-driven decisions that can improve patient outcomes. Given the wealth of data available with the widespread adoption of electronic health records, more flexible statistical models are required that can account for the messiness and complexity of this data. In this dissertation we focus on developing models for clinical time series, as most data within healthcare is collected longitudinally and it is important to take this structure into account. Models built off of Gaussian processes are natural in this setting of irregularly sampled, noisy time series with many missing values. In addition, they have the added benefit of accounting for and quantifying uncertainty, which can be extremely useful in medical decision making. In this dissertation, we develop new Gaussian process-based models for medical time series along with associated algorithms for efficient inference on large-scale electronic health records data. We apply these models to several real healthcare applications, using local data obtained from the Duke University healthcare system.

In Chapter 1 we give a brief overview of clinical prediction models, electronic health records, and Gaussian processes. In Chapter 2, we develop several Gaussian process models for clinical time series in the context of chronic kidney disease management. We show how our proposed joint model for longitudinal and time-to-event data and model for multivariate time series can make accurate predictions about a patient's future disease trajectory. In Chapter 3, we combine multi-output Gaussian

processes with a downstream black-box deep recurrent neural network model from deep learning. We apply this modeling framework to clinical time series to improve early detection of sepsis among patients in the hospital, and show that the Gaussian process preprocessing layer both allows for uncertainty quantification and acts as a form of data augmentation to reduce overfitting. In Chapter 4, we again use multi-output Gaussian processes as a preprocessing layer in model-free deep reinforcement learning. Here the goal is to learn optimal treatments for sepsis given clinical time series and historical treatment decisions taken by clinicians, and we show that the Gaussian process preprocessing layer and use of a recurrent architecture offers improvements over standard deep reinforcement learning methods. We conclude in Chapter 5 with a summary of future areas for work, and a discussion on practical considerations and challenges involved in deploying machine learning models into actual clinical practice.

To the voices in my head.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

t, T	scalars t, T (italics, lowercase and uppercase)
$\mathbf{t}$	vector $\mathbf{t}$ (boldface, lowercase)
$\mathbf{T}$	matrix $\mathbf{T}$ (boldface, uppercase)
$\mathbb{E}$	Expectation
$\mathcal{GP}(m(t), k(t, t'))$	Gaussian Process
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normal distribution

Abbreviations

ACO	Accountable Care Organization
AI	Artificial Intelligence
AMI	Acute Myocardial Infarction (heart attack)
CKD	Chronic Kidney Disease
CMS	Centers for Medicare and Medicaid Services
CVA	Cerebrovascular Accident (stroke)
EGDT	Early Goal Directed Therapy
EHR	Electronic Health Record
EM	Expectation Maximization
GP	Gaussian Process
ICD	International Classification of Diseases

ICU	Intensive Care Unit
LSTM	Long Short Term Memory
MCMC	Markov Chain Monte Carlo
MDP	Markov decision process
ML	Machine Learning
MGP	Multi-output Gaussian Process
OU	Ornstein-Uhlenbeck (covariance function)
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SE	Squared Exponential (covariance function)
SSC	Surviving Sepsis Campaign

Acknowledgements

First, I would like to thank my thesis advisor, Katherine Heller, for her guidance and advice throughout my graduate studies at Duke. Her enthusiasm for tackling hard problems in the machine learning for healthcare space has been inspiring, and I am grateful for her support and for introducing me to important and meaningful collaborations with clinicians. Thanks also to David Dunson for his mentorship, and for many stimulating conversations and innovative ideas. His passion for scholarship and constructive criticism have undoubtedly made me a better researcher. I would also like to thank Joseph Lucas for initially introducing me to electronic health records data and for valuable feedback throughout my studies, and Jerry Reiter for serving on my committee and providing helpful comments.

I have been extremely fortunate to have worked alongside incredible clinical collaborators throughout my time at Duke, especially Cara O'Brien, Armando Bedoya, Meredith Clement, and Blake Cameron. Working alongside such impassioned and tireless physicians who are so eager to use technology to improve patient care has been truly inspirational. Thanks to Mark, Suresh, Nathan, Anthony, and others at the Duke Institute for Health Innovation for facilitating such fruitful collaborations.

I am very grateful to my undergraduate advisor Dan Rockmore and to Nick Foti for introducing me to the wonderful world of machine learning and statistics, starting me down my current academic path. I also owe a great deal to my Duke classmates for their friendship and unwavering support: Victor, Ken, Lorin, Derek,

Matt, Christoph, Maggie. Without you all I would not have made it this far, and you certainly made The Cave an interesting place to work. Thanks as well to other Duke friends for their support and companionship over the years.

Finally, thanks to my loving family for their constant support throughout my life. Special thanks to my parents David and Susan for always pushing me to live up to my full potential. Lastly, my eternal gratitude to Morgan Simons for her endless encouragement, patience, and optimism.

This research has been supported by the Office of Naval Research through the National Defense Science and Engineering Graduate Research Fellowship, as well as by the Statistical and Applied Mathematical Sciences Institute.

1

Introduction

> It appears to me a most excellent thing for the physician to cultivate Prognosis; for by foreseeing and foretelling, in the presence of the sick, the present, the past, and the future, and explaining the omissions which patients have been guilty of, he will be the more readily believed to be acquainted with the circumstances of the sick; so that men will have confidence to intrust themselves to such a physician.
>
> *Hippocrates, 400 BCE*
> *The Book of Prognostics*

1.1 Clinical Prediction Models

Prognosis is fundamental to the practice of medicine, and has been since the time of Hippocrates. Physicians constantly make predictions when they treat patients, and these predictions guide the decisions they make in order to improve prognosis. They may predict things such as the likelihood that a high-risk patient has a disease, whether this patient would benefit from an additional invasive diagnostic test, and ultimately what treatments are most likely to improve the patient's outcome. However, physicians tend to have difficulty in estimating risks of diseases and often err towards overestimation (Friedmann et al., 1996). Historically medicine itself used to

be more subjective, and the underlying probabilities guiding clinical decision-making were implicit, based off of a provider's training and personal experiences treating a limited number of patients. This has changed in the current era of "evidence-based medicine", where the goal is to integrate clinical expertise with the current best external evidence (Sackett et al., 1996). Clinical prediction models offer to provide additional objective evidence to aid physicians in the decision-making process, and might be integrated into workflows through clinical decision support systems (Kawamoto et al., 2005).

There are many areas where prediction models can provide useful knowledge to help inform clinical decisions (Steyerberg, 2009). A prediction model might aid in *screening* for early signs of a disease. This would allow providers to allocate more resources to patients deemed highest risk of developing the disease in the near future, and ensure these patients receive any necessary preventative care. Prediction models can also be created to improve *diagnosis* of diseases or *early detection* of adverse events. Such a model might automatically alert a physician if a patient appears to be at high risk of rapidly deteriorating, allowing for faster treatment and potentially improving their outcome. A prediction model may also be used in enhancing *treatment* and therapy decisions by predicting the efficacy of existing treatments and helping the clinician weigh the estimated benefit of each treatment alongside any side effects or other costs.

In this dissertation we focus on developing statistical models capable of making personalized predictions to help solve clinical problems in all three of these areas, with an emphasis on modeling medical time series. There are many other applications throughout healthcare where prediction models can also provide immense value, especially in operational sectors of hospital management. Though we will not discuss them in this dissertation, these include important problems such as forecasting bed utilization (Barnes et al., 2015) and operating room case volume (Eijkemans et al.,

2010), and predicting which patients are at high risk of being readmitted (Futoma et al., 2015) or of missing scheduled appointments (Huang and Hanauer, 2014).

Clinical prediction models are by no means a recent idea, and have appeared under the name "clinical prediction rules" for some time (Wasson et al., 1985). Many such prediction rules already exist and are in wide use throughout a variety of areas in medicine. A few well known scores include the CHADS2 score for estimating risk of stroke (Gage et al., 2001), the Framingham risk score for estimating risk of developing cardiovascular disease in 10 years (Wilson et al., 1998), the APACHE II score for measuring severity of disease for patients in the intensive care unit (ICU) (Knaus et al., 1985), and the CURB-65 score for assessing severity of pneumonia (Lim et al., 2003).

However, these scores typically suffer from several drawbacks. They are usually based on only a few demographic, vital sign, or laboratory result variables, and assign independent scores to each value using pre-specified thresholds. Although this makes them easy to compute and reason with, it also makes them overly simple and can limit their accuracy. Importantly, these scores generally ignore potentially complex correlations between variables and also ignore their evolution in time, precluding the ability to make dynamic and personalized predictions that can update as more information is collected. Lastly, these prediction rules generally come without any quantifiable notion of uncertainty. This motivates the need to develop more flexible statistical models capable of alleviating these issues, along with efficient inference algorithms that will enable fitting these models to large-scale medical datasets. To this end, we will turn to techniques from machine learning (ML), a subfield of computer science closely related to artificial intelligence (AI) and computational statistics.

In the past decade, there has been a surge of interest in applying machine learning to problems in medicine and healthcare. Figure 1.1 shows the relative number of papers indexed in PubMed on machine learning from 1996 to 2016, indicating

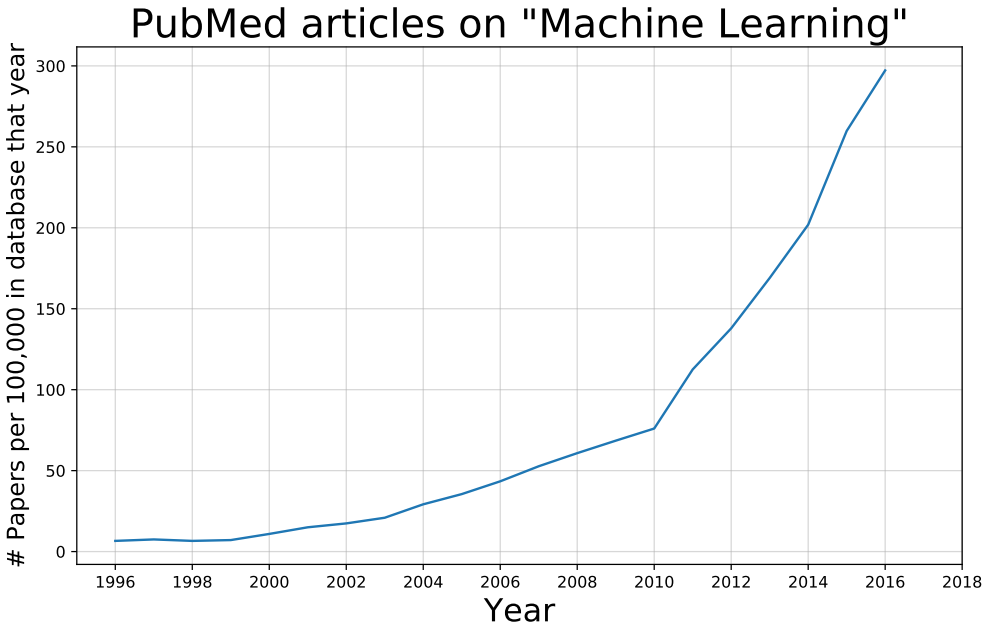

FIGURE 1.1: The relative number of papers indexed in PubMed on "machine learning". The absolute number rose from 30 in 1996 to 3745 in 2016.

this trend. Over the past few years, many articles have emphasized the crucial role that ML and AI will play in medicine in the future (Obermeyer and Emanuel, 2016; Darcy et al., 2016; Beam and Kohane, 2016). This is due in part to breakthroughs in speech recognition, computer vision, and textual understanding in ML that are now integrated into many products and services in widespread use. Recent prominent successes in applying ML to healthcare largely center around the use of deep learning to solve challenging problems in medical imaging, such as classifying skin cancer (Esteva et al., 2017), detecting lymph node metastases (Bejnordi et al., 2017), identifying diabetic retinopathy (Gulshan et al., 2016; Ting et al., 2017), and predicting respiratory events and chronic obstructive pulmonary disease stage (González et al., 2018).

However, perhaps the most important factor driving interest in machine learning for healthcare is the massive amounts of medical data being automatically collected

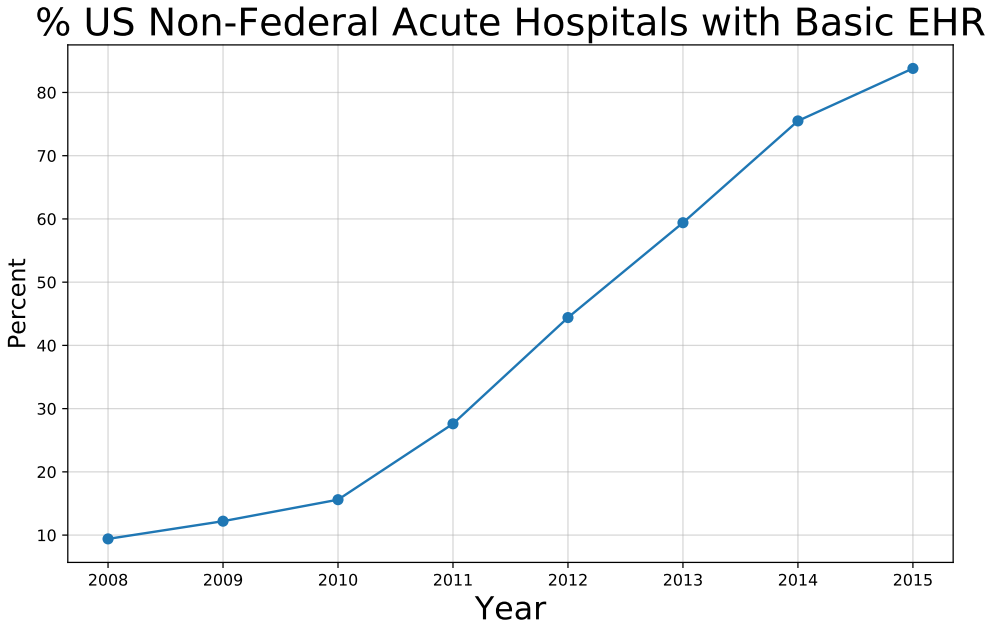

FIGURE 1.2: Percent of non-federal acute care US Hospitals with adoption of at least a Basic EHR with notes system and 10 core functionalities deemed essential.

during routine clinical care, due to the widespread proliferation of electronic health records (EHRs). Figure 1.2 shows the rapid increase in EHR adoption over the past decade (Henry et al., 2016). In this era of "Big Data", EHRs contain unprecedented quantities of healthcare data that can generate actionable knowledge and value for patients, clinicians, administrators, researchers, and health policy makers alike (Weil, 2014; Roski et al., 2014; Krumholz, 2014).

1.2 Electronic Health Records Data

In this section, we provide some background on EHRs; see Goldstein et al. (2017) for a recent systematic review of many risk prediction models built from EHR data.

Throughout this dissertation, all of our datasets come directly from the Duke University Health System's EHR (Epic Systems, Verona, WI). In general, an EHR stores nearly all available information captured about patients during their encoun-

ters within a health system. As such, it contains a large quantity of longitudinal patient data. The vast majority of the data are unstructured, contained within free-text notes and reports. Structured data include demographics, diagnosis and procedural codes, medication orders, laboratory results, and other objective clinical observations (such as vital signs and various nursing assessments).

The EHR stores granular information about medical diagnoses using structured, hierarchical codes. In the US, prior to October 1, 2015 these codes conformed to ICD-9-CM (International Classification of Diseases, 9th revision, Clinical Modification), and afterwards conformed to ICD-10-CM (International Classification of Diseases, 10th revision, Clinical Modification). These are differing versions of a standardized taxonomy of codes that are used principally for medical billing. For each medical encounter (such as a clinic or emergency department visit), a set of codes is assigned to document the primary problems or diseases that were addressed. In total, there are about 13,000 unique ICD-9 codes, and 68,000 ICD-10 codes. Each clinical diagnosis may have multiple corresponding ICD diagnosis codes. The Agency for Healthcare Research and Quality publishes the Clinical Classifications Software[1], a categorization tool that collapses the thousands of original codes into a few hundred clinically meaningful concepts.

In contrast to diagnosis codes, which capture clinicians' subjective diagnostic impressions, laboratory tests provide objective clinical data. A single medical encounter may include dozens, hundreds or (in the case of hospitalizations) thousands of discrete laboratory test results. Identifying and grouping relevant laboratory test results can be difficult due to lack of standardization, changing conventions over time, and idiosyncratic factors native to one health system. For example, serum creatinine, which is a lab test often used to monitor kidney function, has more than 18 different names in the Duke EHR that refer to the same value (e.g. "CREA",

[1] `http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp`

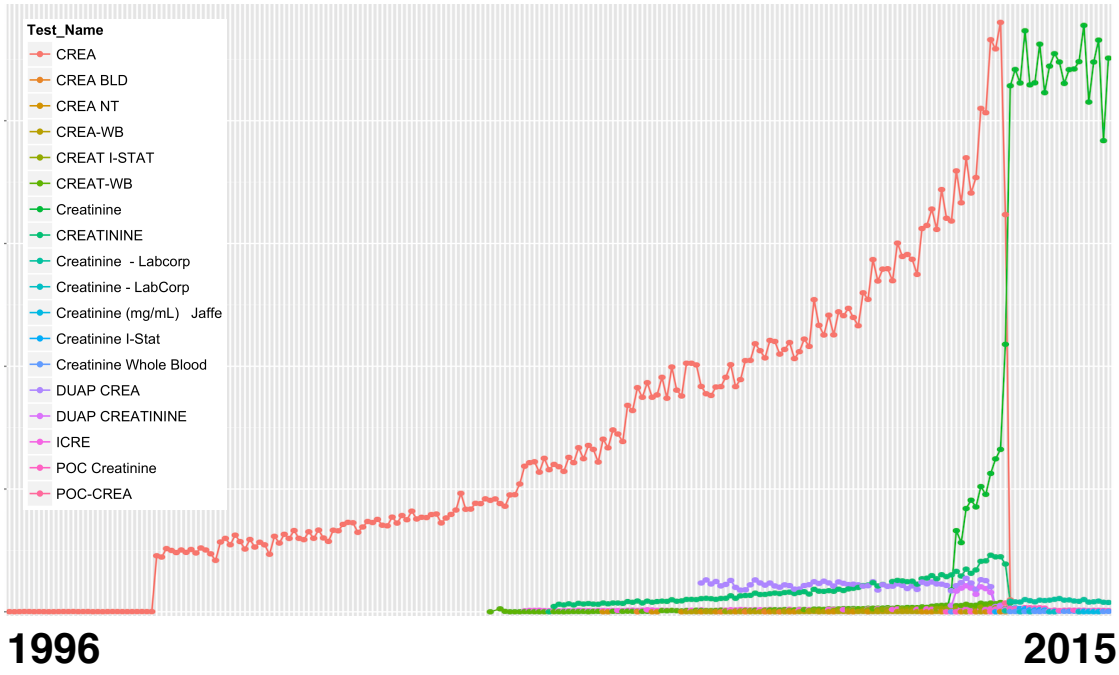

FIGURE 1.3: Counts of different identifiers for serum creatinine each month from 1996 to 2015 in the Duke University health system EHR. Note the abrupt change in 2013, when a legacy EHR was replaced by one developed by Epic Systems.

"Creatinine", "DUAP CREA"). Figure 1.3 shows how often each of the different names appears over time in our EHR. Harmonizing and grouping these lab results required an exhaustive review of laboratory metadata by a subject matter expert. This type of manual data cleaning is common when beginning a new analysis on a different dataset.

Despite the wealth of information contained in EHRs, there are numerous limitations that can cause practical challenges in implementing clinical prediction models using them (Hersh et al., 2013; Amarasingham et al., 2014). Data quality is often poor, complicated by inaccurate, inconsistent or missing information. The EHR at a single organization may fail to capture the full patient story and all relevant outcomes of interest, as is the case when patients receive care from multiple, non-interoperable healthcare systems over time. Relevant patient reported outcomes,

such as perceived quality of life, are rarely captured by EHRs. Events such as death may not be registered, particularly when patients die outside of the hospital. Much of the information about a patient may be buried in clinical notes (in the form of free text), and may not be recoverable. Though this source of data can be rich in information, it is significantly harder to automatically extract meaningful features from medical text.

Even when all of the relevant data elements needed to build a predictive model have been identified, extracted, and cleaned, there are still subtle issues to be wary of. Many clinical data are collected for billing purposes rather than patient care or research, distorting the relative importance of certain elements. Changes in coding practices may imply clinical differences where they do not exist. Data may be biased, and are frequently missing not at random (Sterne et al., 2009; Weiskopf et al., 2013). Certain laboratory tests may be performed only when a clinician suspects an abnormality, and a diagnosis is only ever recorded when a disease is suspected. If a patient has a test done and the result is negative, the mere fact that a test was even performed contains valuable information. Finally, all data collected is observational, so treatments given are not randomized. Treatments can be confounded by clinical factors unrelated to the condition being treated, e.g. patients who are overall sicker with other comorbidities. Analysis involving estimation of treatment effects requires methods from causal inference to control for these biases (Holland, 1986; Imbens and Rubin, 2015).

Despite these all of these potential issues, EHRs still offer a rich data source that can be used to tackle meaningful problems in healthcare. However, in order to do so we need to account for the structure we expect in the data, and make our assumptions explicit. In this dissertation, we are primarily interested in modeling longitudinal data and clinical time series, and the EHR contains a variety of longitudinal data on many different time scales. Data collected at outpatient clinics for management

of chronic diseases may span long periods of times, on the timescale on months to years. Data collected from an inpatient setting during a single hospitalization will be more granular, on a timescale of hours to minutes. Regardless of the particular application, medical data is collected over time, and treatment decisions are made in the context of a patient's previous medical history. Models that account for this temporal structure and use information about both a patient's current physiological state and history of past states tend to perform better on the types of tasks we are interested in.

The models we develop in this dissertation all build off of Gaussian Processes (GPs), a foundational class of Bayesian nonparametric statistical models well suited to modeling irregularly sampled time series, among other things. GPs provide a principled, practical, and probabilistic approach to building models, and inherit many of the attractive properties of the Bayesian modeling framework, such as more precise uncertainty quantification and the ability to combine prior information with data in a solid decision theoretic framework (Berger, 2013).

1.3 Gaussian Processes

In this section we provide a short introduction to Gaussian processes. In Bayesian modeling, GPs are commonly used as prior distributions for functions. In the context of irregularly sampled time series, a GP can serve as a prior distribution for a latent function f defined over the real line, representing the unknown "true" value of a quantity of interest over time (e.g. a biomarker indicative of disease severity). Although we emphasize their use in time series modeling where typically the input space is only a single dimension ($\mathbb{R}$), they can also be defined on arbitrary input spaces. A more complete background on GPs can be found in (Rasmussen and Williams, 2005).

1.3.1 Definition and Model

Let $\mathcal{I}$ denote an arbitrary index set which may be finite, countably infinite, or uncountably infinite (e.g. in time series applications, we typically have $\mathcal{I} = \mathbb{R}$). We say that the collection of random variables $\{f_i\}_{i \in \mathcal{I}}$ form a *Gaussian process* if, for any finite subset $\mathcal{I}' \subset \mathcal{I}$, the random variables $\{f_i\}_{i \in \mathcal{I}'}$ have a joint Gaussian distribution. That is, a Gaussian process is a collection of random variables, any finite number of which have a joint multivariate normal distribution.

A Gaussian process is completely specified in terms of a mean function and a covariance function. In this exposition, we assume that our input space is univariate (i.e. that the GP is defined over $\mathbb{R}$), but they can also be applied to more general spaces. We define the mean function $m(t)$ and covariance function (also referred to a kernel function) $k(t, t')$ of the process $f(t)$ as

$$m(t) = \mathbb{E}[f(t)] \tag{1.1}$$

$$k(t, t') = \text{cov}(f(t), f(t')) = \mathbb{E}[(f(t) - m(t))(f(t') - m(t'))], \tag{1.2}$$

and denote the Gaussian process by

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')). \tag{1.3}$$

Given a finite collection of inputs $\mathbf{t} = [t_1, \ldots, t_n]^\top$, then the associated set of function values $f(\mathbf{t}) \in \mathbb{R}^n$ is jointly Gaussian distributed, with explicit form

$$f(\mathbf{t}) \sim \mathcal{N}(m(\mathbf{t}), K(\mathbf{t}, \mathbf{t})). \tag{1.4}$$

Here $m(\mathbf{t}) \in \mathbb{R}^n$ denotes the vector of values of the mean function evaluated at the inputs $\mathbf{t}$; $K(\mathbf{t}, \mathbf{t}) \equiv \mathbf{K}_{\mathbf{t},\mathbf{t}} \in \mathbb{R}^{n \times n}$ denotes the full covariance matrix of all function values $f(\mathbf{t})$, as specified by evaluating the covariance function k between all input pairs, $[K(\mathbf{t}, \mathbf{t})]_{ij} = k(t_i, t_j)$; and $\mathcal{N}$ denotes the multivariate normal distribution. We will use similar notation throughout the remainder of this dissertation. For now we

make the common simplifying assumption that the prior mean is zero, i.e. $m(t) = 0$, although this can easily be relaxed. Note that even though Gaussian processes are fundamentally infinite-dimensional objects, in practice most manipulations with them only involve computations with multivariate Gaussian densities, as seen here.

GPs are often used as a nonparametric regression model. In the context of time series modeling, assume we have data $\mathcal{D} = \{\mathbf{t} = [t_1, \ldots, t_n]^\top, \mathbf{y} = [y_1, \ldots, y_n]^\top\}$ generated by $y_i = f(t_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is independent Gaussian measurement noise and f is an unknown regression function. We can consider f as a random function, and infer a posterior distribution $p(f|\mathcal{D})$ over possible functions f from the GP prior $p(f)$, the data $\mathcal{D}$, and high-level assumptions about the smoothness of f encoded in the covariance function.

1.3.2 Posterior Inference and Prediction

We use Bayesian inference in order to find a posterior distribution of the random function f. In Bayesian inference, we fit a probability model to a set of data and summarize our results in terms of a probability distribution on the unknown quantities of interest, in this case f (Gelman et al., 2013).

Having observed $y_i = f(t_i) + \epsilon_i$, $i = 1, \ldots, n$, from Bayes' Theorem we have that

$$p(f|\mathbf{t}, \mathbf{y}, \theta) = \frac{p(\mathbf{y}|f, \mathbf{t}, \theta)p(f|\theta)}{p(\mathbf{y}|\mathbf{t}, \theta)}, \tag{1.5}$$

where θ is any hyperparameters associated with the covariance function k or the mean function m. In assuming independent Gaussian measurement noise, we have that the observations y_i are conditionally independent given f. Thus, the likelihood factorizes:

$$p(\mathbf{y}|f, \mathbf{t}, \theta) = \prod_{i=1}^{n} p(y_i|f(t_i)) = \prod_{i=1}^{n} \mathcal{N}(y_i|f(t_i), \sigma^2) = \mathcal{N}(\mathbf{y}|f(\mathbf{t}), \sigma^2\mathbf{I}). \tag{1.6}$$

Combining the likelihood in equation (1.6) and the GP prior in equation (1.3) results in the posterior distribution in equation (1.5). Since both terms are Gaussian, it is easy to show that the resulting posterior is also a GP, with mean and covariance functions given by:

$$m_{post}(t) \equiv \mathbb{E}[f(t)|\mathbf{t}, \mathbf{y}, \theta] = \mathbf{k}_{t,\mathbf{t}}(\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{1.7}$$

$$k_{post}(t, t') \equiv \text{cov}[f(t), f(t')|\mathbf{t}, \mathbf{y}, \theta] = k(t, t') - \mathbf{k}_{t,\mathbf{t}}(\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{t},t'}, \tag{1.8}$$

where $\mathbf{k}_{t,\mathbf{t}} \equiv [k(t, t_1), \ldots, k(t, t_n)] \in \mathbb{R}^n$ denotes a vector with covariances between a test input t and all training times $\mathbf{t}$, and $\mathbf{k}_{t,\mathbf{t}} = \mathbf{k}_{\mathbf{t},t}^\top$. More concretely, given a vector of testing inputs $\mathbf{t}^*$, then the posterior predictive distribution for $f(\mathbf{t}^*)$ is:

$$f(\mathbf{t}^*)|\mathbf{t}, \mathbf{y}, \theta \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{1.9}$$

$$\boldsymbol{\mu}^* = \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}(\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{1.10}$$

$$\boldsymbol{\Sigma}^* = \mathbf{K}_{\mathbf{t}^*,\mathbf{t}^*} - \mathbf{K}_{\mathbf{t}^*,\mathbf{t}}(\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{t},\mathbf{t}^*}. \tag{1.11}$$

While $\boldsymbol{\mu}^*$ can be used as a point estimate, we have a full distribution for $f(\mathbf{t}^*)$, allowing us to quantify the associated uncertainty in our predictions.

1.3.3 Examples of GP Regression

We give a brief example of a GP regression, using two common covariance functions. The choice of covariance function is an important modeling decision, as different covariance functions will endow the resulting GP with different properties. We present both functions in a normalized form, with $k(0, 0) = 1$; we can multiply k by a positive constant σ_f^2 to get any desired process variance. Both covariance functions we consider are stationary and isotropic, meaning $k(t, t')$ only depends on $|t - t'|$. Informally, this means that the covariance structure of the resulting process only depends on the distance between points and not on their actual location.

The first covariance function is the squared exponential (SE) covariance function

$$k_{SE}(t, t') = \exp\left(-\frac{(t - t')^2}{l}\right), \tag{1.12}$$

where l is a length-scale parameter, informally describing how "wiggly" functions should be. This covariance function is infinitely differentiable, meaning that a GP with this covariance function has mean square derivatives of all orders and will be very smooth.

The second covariance function we consider is the Ornstein-Uhlenbeck (OU) covariance function

$$k_{OU}(t, t') = \exp\left(-\frac{|t - t'|}{l}\right), \tag{1.13}$$

where again l is a length-scale. This covariance function is also a special case of the more general Matérn family of covariance functions (with parameter $\nu = 1/2$; the SE is also a special case and arises in the limit $\nu \to \infty$). A resulting GP with this covariance function will be mean square continuous but not differentiable. The name arises as it is the covariance function of the OU process used to model the velocity of a particle undergoing Brownian motion. Interestingly, the OU covariance function gives rise to a particular type of continuous-time $AR(1)$ autoregressive GP; see Rasmussen and Williams (2005) for more details.

It is common practice to assume that the prior mean function is zero, i.e. $m(t) = 0$, so we will do so in this example. In Figure 1.4 we show 4 draws each from a zero-mean GP with SE and OU covariance functions, both with $l = 2$ and unit variance $\sigma_f^2 = 1$. Clearly the SE kernel yields very smooth functions, while the OU kernel creates rough functions. Next, we sample 10 data points, and show the resulting posterior distribution for f in a GP regression model, using each of these GPs (we fix $\sigma = .05$ so that there is very low noise). Figure 1.5 shows the data, resulting mean function for both posterior GPs, and 3 draws from the resulting posterior for each process. In areas far from the data, uncertainty increases substantially and the process reverts to the mean of 0. Around the data, uncertainty on f is considerable lower. And, again the GP with SE kernel is considerably smoother.

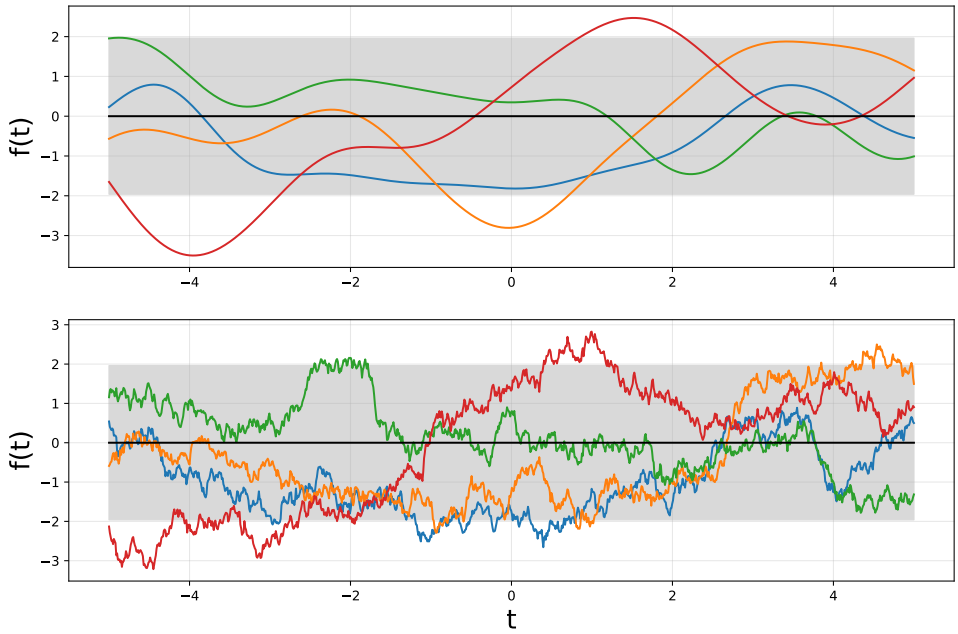

FIGURE 1.4: Draws from a zero mean GP prior. Black line denotes zero mean function, and shaded region denotes 95% credible interval. Top: 4 draws from a GP with SE covariance function. Bottom: 4 draws from a GP with OU covariance function.

1.3.4 Hyperparameter Learning

So far, we have assumed that any hyperparameters θ to the covariance function are fixed and known. Sometimes there is strong prior knowledge about the function shapes we expect, and this is reasonable. In many situations, however, we may want to learn θ. This is straightforward to do in a fully Bayesian setup. We place a hyper-prior $p(\theta)$ on their values, and can write the marginal posterior on hyperparameters after integrating out f as

$$p(\theta|\mathbf{t}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{t}, \theta)p(\theta)}{p(\mathbf{y}|\mathbf{t})}, \tag{1.14}$$

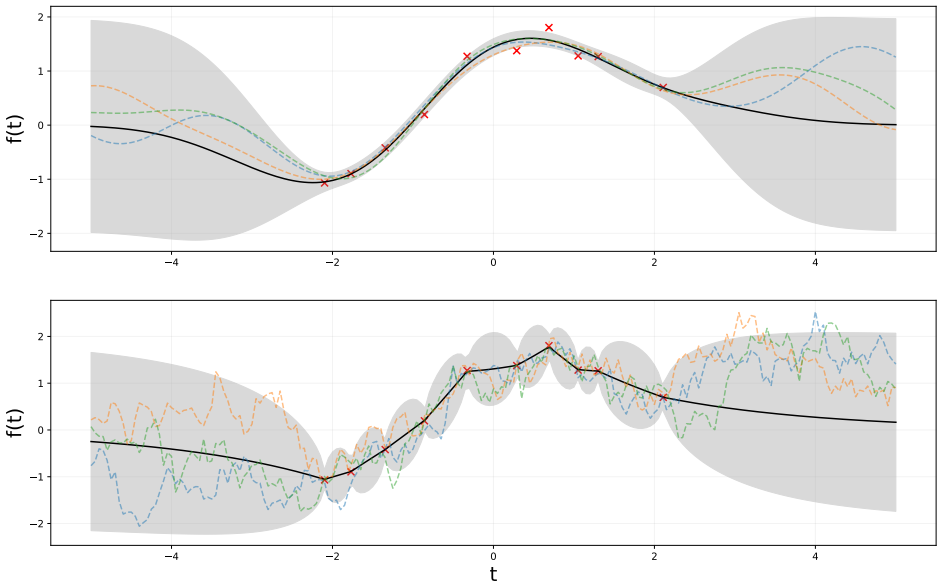

FIGURE 1.5: Top: Posterior GP with SE covariance function. Bottom: Posterior GP with OU covariance function. 10 data points are shown as red x's. 3 draws from the posterior are shown, along with the mean function in black and 95% point-wise credible intervals in grey.

where the likelihood term for the hyperparemeters $p(\mathbf{y}|\mathbf{t}, \theta)$ given the data, after marginalizing out f, is the *marginal likelihood*, and also the normalizing constant from Equation 1.5:

$$p(\mathbf{y}|\mathbf{t}, \theta) = \int p(\mathbf{y}|\mathbf{t}, f, \theta)p(f|\theta)df. \tag{1.15}$$

The marginal likelihood has a closed form in standard GP regression since the prior and likelihood are both Gaussian, and is given by:

$$\log\, p(\mathbf{y}|\mathbf{t}, \theta) = -\frac{1}{2}\mathbf{y}^\top(\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log\, |\mathbf{K}_{\mathbf{t},\mathbf{t}} + \sigma^2\mathbf{I}| - \frac{n}{2}\log\, 2\pi. \tag{1.16}$$

This is a complex function of θ (θ appears implicitly through the covariance matrix $\mathbf{K}_{\mathbf{t},\mathbf{t}}$), meaning the hyperparameter posterior in Equation 1.14 will clearly not be analytically tractable. Approximate computation of the hyperparameter poste-

rior would generally require some sort of computationally demanding Monte Carlo method. In this dissertation, we will not follow this fully Bayesian path to the end, and instead will find good point estimates $\hat{\theta}$ for the hyperparameters on which we condition our inference. This can be done in this simple GP regression model by finding a θ that maximizes Equation 1.14 to obtain a maximum a posteriori (MAP) estimate. In the case where we choose a "flat" hyper-prior $p(\theta)$, assuming any values for the hyperparameters are possible, then the MAP estimate for θ coincides with the maximum (marginal) likelihood estimate, obtained by maximizing Equation 1.16. Often this maximization is done using gradient-based methods. This problem can be hard in some cases, as the optimization problem is nonlinear and non-convex.

1.3.5 Approximate Methods

There is a substantial computational overhead to working with GPs. The bottleneck is generally computing $(\mathbf{K_{t,t}} + \sigma^2\mathbf{I})^{-1}$ and $\log |\mathbf{K_{t,t}} + \sigma^2\mathbf{I}|$ that appear in the predictive distributions and marginal likelihood, where $\mathbf{K_{t,t}}$ is the $n \times n$ covariance matrix between training inputs. Usually this is accomplished using the Cholesky decomposition of $\mathbf{K_{t,t}}$, which incurs $\mathcal{O}(n^3)$ computations and $\mathcal{O}(n^2)$ storage, and afterwards predictions cost $\mathcal{O}(n^2)$ per test point. For this reason, GPs are often limited to a few thousand training points at most.

Many promising approaches to scalability have been explored to alleviate these issues. A common strategy is to exploit low-rank approximations for the kernel matrix; the resulting models are often referred to as sparse GPs. This was first done in Silverman (1985), and was further improved upon in Snelson and Ghahramani (2006); Quiñonero-Candela and Rasmussen (2005) provides a unifying view of this line of work. The main idea is to augment the GP prior $p(\mathbf{f}, \mathbf{f}^*)$ of the latent function at the training data, $\mathbf{f}$, and at the test data, $\mathbf{f}^*$, with an additional set of latent variables $\mathbf{u} = [u_1, \ldots, u_m]$, referred to as *inducing variables*, where generally $m << n$.

These are values of the GP at an additional set of input locations $\mathbf{t_u}$, called *inducing inputs*. The original prior $p(\mathbf{f}, \mathbf{f}^*)$ can be recovered by marginalizing $\mathbf{u}$ out of the joint GP prior, $p(\mathbf{f}, \mathbf{f}^*) = \int p(\mathbf{f}, \mathbf{f}^*|\mathbf{u})p(\mathbf{u})d\mathbf{u}$. The approximation arises by assuming $\mathbf{f}$ and $\mathbf{f}^*$ are conditionally independent given $\mathbf{u}$, i.e.

$$p(\mathbf{f}, \mathbf{f}^*) \approx q(\mathbf{f}, \mathbf{f}^*) = \int q(\mathbf{f}^*|\mathbf{u})q(\mathbf{f}|\mathbf{u})p(\mathbf{u})d\mathbf{u}, \tag{1.17}$$

with different approaches making different assumptions on the conditional distributions q and on how the inducing variables are chosen. It is possible to treat the inducing inputs $\mathbf{t_u}$ as hyperparameters and optimize their location, although this can lead to overfitting. The number of inducing variables directly determines how faithful the approximation is, and the resulting model will have computational complexity $\mathcal{O}(m^2 n)$.

More recent approaches turn to stochastic variational inference methods instead. These techniques are significantly more scalable since they rely on stochastic gradient descent, and only process the data in smaller subsamples. In variational inference, a lower bound on the true log marginal likelihood is maximized, which is equivalent to minimizing the Kullback-Leibler divergence between a variational approximation to the posterior and the exact posterior distribution over latent function values. Titsias (2009) presented the first variational approach to learning a sparse GP model, where treating the inducing inputs as variational parameters reduced overfitting. This approach was extended by Hensman et al. (2013), who derive a stochastic variational GP posterior over the inducing variables that properly factorizes, so that stochastic gradient descent can be applied. This is noteworthy, as normally the GP marginal likelihood does not admit a factorization over individual data points, precluding the use of stochastic optimization methods. The overall complexity of this approach is now only $\mathcal{O}(m^3)$, with no direct dependence on the data size, allowing scaling to large n. However, m may still need to be large in some cases in order to increase the

accuracy of the approximation. This idea has since been further refined to handle non-conjugate models, where sampling methods are used to compute non-conjugate expectations (Dezfouli and Bonilla, 2015; Hensman et al., 2015a,b), allowing the use of large-scale GPs in more sophisticated models. Development of GP models that are both flexible and scalable is still a very active field of research.

1.4 Outline

Having provided some background on clinical prediction models, machine learning in healthcare, EHRs, and GPs, we now outline the remainder of this dissertation.

In Chapter 2, we present several novel Gaussian process-based models for clinical time series in the context of chronic disease management. We show how a flexible GP-based model for univariate time series can be extended to also jointly model time-to-event data, or to instead model multivariate time series. Work in this chapter previously appeared in Futoma et al. (2016b) and Futoma et al. (2016a), and was in collaboration with Mark Sendak, Blake Cameron, and Katherine Heller.

In Chapter 3, we present a novel methodology for tying together multi-output Gaussian processes for multivariate time series to deep recurrent neural networks from deep learning. We apply this model to early detection of sepsis, and show how this approach offers improved predictive performance over standard deep learning methods for time series data, as well as other baseline methods and clinical scores. Work in this chapter previously appeared in Futoma et al. (2017b) and Futoma et al. (2017a), and was in collaboration with Sanjay Hariharan, Mark Sendak, Nathan Brajer, Armando Bedoya, Meredith Clement, Cara O'Brien, and Katherine Heller.

In Chapter 4, we present preliminary work in progress, and transition from predictive to prescriptive modeling. We apply the Gaussian Process adapter from Chapter 3 to a reinforcement learning (RL) framework, where the goal is to learn an optimal treatment regime for improving treatment of sepsis. We show that use of the

GP adapter leads to a higher value policy with potentially lower mortality than the physician policy used to generate the data. Exploratory analyses suggest that the GP adapter acts as a data augmentation technique that helps stabilize the learning. A preliminary version of this work appeared in Futoma et al. (2017c), and was in collaboration with Mark Sendak, Anthony Lin, Armando Bedoya, Meredith Clement, Cara O'Brien, and Katherine Heller.

Finally, we conclude in Chapter 5 with a summary of contributions of this work, along with some final considerations on areas for future research to aid in the deployment of machine learning into real clinical practice.

2

Gaussian Process Models for Chronic Kidney Disease Management

2.1 Introduction

With the dawn of precision medicine, accountable care, and alternative payment models it will become increasingly important for healthcare organizations to make accurate predictions about individual patients' future health risks to improve quality and contain costs (Centers for Medicare and Medicaid Services, 2016). In particular, managing patients with complex chronic diseases such as cardiovascular disease, diabetes, and chronic kidney disease is especially difficult, as selection of optimal therapy may require integration of multiple conditions and risk factors that are considered in isolation under current approaches to care. Further, patients with multiple chronic conditions are among both the most expensive and highest utilizers of healthcare services (Johnson et al., 2015).

Accountable care organizations (ACOs) are organizations that bear financial responsibility for the quality and total cost of healthcare services provided to a defined population of patients. In order to deliver the right care at the right time in the right

setting, ACOs need personalized prediction tools that identify individual patients in their populations at greatest risk of having poor clinical outcomes (Parikh et al., 2016; Bates et al., 2014). Most ACOs currently lack these capabilities (Leventhal, 2016). However, with the widespread adoption of electronic health records (EHRs), much of the data necessary to build such tools are already being collected during the course of routine medical care. In order to be clinically useful, such tools should be flexible enough (1) to accommodate the limitations inherent to operational EHR data (Hersh et al., 2013); (2) to update predictions dynamically as new information becomes available; and (3) to scale to the massive size of modern health records.

We collaborated with Duke Connected Care, the ACO affiliated with the Duke University health system, to develop predictive tools for chronic kidney disease (CKD). CKD is characterized by a gradual and generally symptomless loss of kidney function over time. CKD and its complications cause poor health, premature death, increased health service utilization, and excess economic costs. CKD is defined and staged by the degree to which a person's estimated glomerular filtration rate (eGFR) is impaired. eGFR is an approximation of overall kidney function and is calculated using a routinely obtained clinical laboratory test (serum creatinine or cystatin C) and demographic information (age, sex and race) (Levey et al., 2009; Kidney Disease: Improving Global Outcomes CKD-MBD Work Group, 2009). Most clinical laboratories report eGFR automatically with every serum creatinine measurement. In addition, a number of other routinely measured laboratory values may be used to help detect abnormal or declining kidney function.

Healthcare providers struggle at many levels to provide optimal care for patients with CKD. First, the majority of healthcare providers fail to recognize the presence of CKD, despite the fact that CKD can be readily identified using simple, eGFR-based laboratory criteria (Szczech et al., 2014; Tuot et al., 2011; Allen et al., 2011). Second, among those patients with recognized CKD, both primary care providers and kidney

specialists struggle to predict which patients will progress to kidney failure (requiring dialysis or kidney transplantation to survive) or suffer from other complications caused by CKD, such as early death from heart attack or stroke (Mendelssohn et al., 2011). Third, providers often fail to prescribe appropriate preventive treatment to slow disease progression or address complications (Smart et al., 2008). Medications such as RAAS drugs can slow progression of CKD if used early enough, while patient counseling and advanced planning can reduce the physical and psychological trauma when kidney failure is imminent.

These traits make CKD an ideal condition to develop disease progression models that can be used in high-impact care management programs. The difficulties with CKD care are best explained with a representative clinical case, illustrated in Figure 2.1. A 47 year-old man makes first contact with the health system for emergency treatment of a stroke. Though he presents with normal kidney function, he possesses several risk factors for developing CKD. Over the next few years, his kidney function rapidly deteriorates (normal annual rate of kidney function loss at his age is only about 1-2 mL/min). However, his kidney disease goes unnoticed due to other more pronounced medical conditions, and he does not receive any treatment to slow progression to total kidney failure. At age 52 he is finally referred to a kidney specialist, more than a year after his kidney function has fallen below the recommended threshold for such a referral. By now, it is too late to make advanced preparations for kidney failure, such as pre-emptive kidney transplantation or at-home dialysis, and kidney failure is inevitable. Within a few months of this first specialist appointment, he requires hospitalization for emergency dialysis initiation, a traumatic procedure that also makes him among the most expensive type of patient to treat (Johnson et al., 2015). While on dialysis over the next decade he suffers several cardiovascular complications before ultimately dying at age 63. A care management program utilizing individualized predictions from a disease progression model, like

FIGURE 2.1: 15-year clinical course of an example patient who experienced both a rapid progression of CKD and a number of other serious health events. Y-axis indicates estimated glomerular filtration rate (eGFR), an estimate of overall kidney function (60-100 is normal, <60 indicates clinically significant kidney disease). X-axis indicates patient age in years. Markers indicate health service use and adverse events. The models we develop allow us to jointly model progression of CKD, as well as the association between the disease progression and risk for adverse events.

the one developed in this work, might have been able to act upon the multiple missed opportunities from this patient story.

However, predicting future disease trajectory is an extremely challenging problem. One difficulty is the many underlying sources of variability that can drive the different potential manifestations of the disease. For instance, the underlying biological mechanisms of the disease can give rise to latent disease subtypes, or groups of individuals with shared characteristics (Saria and Goldenberg, 2015). For most com-

plicated diseases, there are no clear definitions of subtypes, so this must be inferred from the data. In addition, there are individual-specific sources of variability that may not be directly observed, such as behavioral and genetic factors, environmental conditions, or temporary infections. Another challenge is the fact that observations are irregularly sampled, asynchronous, and episodic, precluding the use of many time series methods developed for data regularly sampled at discrete time intervals. The large degree of missing data, especially when modeling multivariate longitudinal data, also presents complications. The task is made even more difficult when the primary data source is the EHR rather than a curated registry, as even selecting a relevant cohort of patients to model can take extensive review by a clinical expert.

2.1.1 Overview of Models

Our aim in this chapter is to develop flexible and broadly applicable statistical models for longitudinal clinical data, in order to provide individualized predictions about the future trajectory of a disease. Two fundamental ideas motivate the development of the two main models developed in this chapter.

The first is that accurate determination of the risk of developing serious complications associated with a disease or its comorbidities may be more clinically useful than prediction of future disease trajectory in some cases. To this end, in the first part of this chapter we propose statistical methods that model both the risks of future loss of kidney function and the risks of future complications or adverse health events. The predictions from these models can then be used by healthcare organizations to connect high-risk patients to appropriately targeted interventions. Since the broad aim is to predict which patients will worsen in the near future, we need to model associations between CKD and the multitude of various health outcomes that could occur. CKD frequently coexists with and contributes to cardiovascular disease. In fact, most patients with advanced CKD pass away from cardiovascular complications

before the onset of kidney failure. In this article, we choose to focus on two common types of adverse cardiovascular events: heart attacks (acute myocardial infarctions [AMIs]) and strokes (cerebrovascular accidents [CVAs]). We will present a Bayesian joint model that flexibly captures the eGFR trajectory of CKD progression, while simultaneously learning the association between disease trajectory and cardiovascular events.

The second idea underpinning the model developed in the second half of the chapter is that for many complex chronic diseases with high heterogeneity, there is not always a single readily available biomarker to quantify disease severity. Additionally, even when such a clinical variable exists, there are often additional related biomarkers that may help improve prediction of future disease state. To this end, in the second half of this chapter we propose a statistical model for multivariate clinical time series that not only accurately models related time-varying clinical variables, but also leverage information between them to improve prediction of the disease trajectory of interest. For instance, in our particular application to CKD, while eGFR is the primary biomarker of interest, prediction of other labs can also be clinically useful.

Both methods we will present in this chapter utilize a hierarchical latent variable model. Each patient is represented by a set of latent variables characterizing both their disease trajectory, and either their risk of having adverse cardiac events (for the joint model) or capturing dependencies between related multivariate longitudinal trajectories of other biomarkers. Both models rely on a Gaussian process (GP) with a highly structured mean function to model each longitudinal variable for each individual. In the joint model, this GP will be tied to an inhomogeneous Poisson process that characterizes the rate of (potentially recurrent) events, while in the multivariate time series model, there will be a different GP for each variable, and the mean functions for the GPs are made dependent through shared latent variables.

Using our models, we study a large cohort of patients with CKD from the Duke

University health system EHR. We make predictions about the future trajectory of their disease severity, as measured by eGFR, along with their risk of cardiovascular events and predictions about five other commonly recorded laboratory values that are known to be affected by CKD. We derive stochastic variational inference algorithms to fit the models that scale well to our large dataset, and both models make accurate predictions that outperform competitive baselines. The model predictions were eventually utilized by our local accountable care organization in a population health rounding tool during chart reviews of high risk patients.

2.2 Related Work

2.2.1 Clinical Prediction Models for Kidney Disease

The vast majority of clinical prediction models developed in the medical literature are cross-sectional, and only consider features at or up until the current time to predict outcomes at a fixed point in the future. These models only attempt to explain variability in the outcome of interest by conditioning on baseline covariates. This precludes the ability to generate dynamic individualized predictions, making them difficult to use for medical decision making in practice. Tangri et al. (2011) is a widely cited Cox proportional hazards model for predicting time to kidney failure, and was one of the first developed renal prediction models. Several review articles provide a summary of clinical prediction models developed for kidney disease, and all are cross-sectional and either logistic or Cox regressions (Tangri et al., 2013; Echouffo-Tcheugui and Kengne, 2012).

2.2.2 Related Machine Learning Models in Healthcare

Within the statistics and machine learning communities, Markov models of many varieties are frequently used to generate dynamic predictions, e.g. autoregressive models, hidden Markov models (HMMs), and state space models (Murphy, 2012). How-

ever, these methods are generally only applicable in settings with discrete, regularly-spaced observation times, and in most applications the data consists of a single set of multivariate time series (e.g. financial returns), not a large collection of irregularly sampled time series as in our setting with disease or lab trajectories. A notable exception is Liu et al. (2015), which applies a continuous-time analogue of HMMs to model disease progression in glaucoma. GPs are much more commonly used in settings with continuous time observations; Roberts et al. (2013) presents a thorough overview. Since they are prior distributions over functions, they are a natural modeling choice for disease trajectories, however, accurate forecasts for GPs require careful specification of the mean functions (Shi et al., 2005).

There are many related works that tackle the problem of dynamic predictions for medical applications. Rizopoulos (2011) construct models with a focus on updating dynamic predictions about time to death as more values of a longitudinal biomarker are observed. They also account for individual heterogeneity using random effects. Related work in this area uses a mixture model to address heterogeneity (Proust-Lima and Taylor, 2009). Other work uses multitask GPs to model multivariate longitudinal clinical data from the Intensive Care Unit (ICU) (Durichen et al., 2015; Ghassemi et al., 2015). However, these works use independently trained models for each patient and do not hierarchically share any information across patients. This worked well in their examples, since in the ICU there is a relatively large number of observations per subject, but would not work as well with our much sparser EHR data for chronic disease patients. On a different note, Lian et al. (2015) use hierarchical point processes to predict hospital admissions, and Ranganath et al. (2015) develop a dynamic factor model to learn relationships between diseases and predict future diagnosis codes. Most similar to our work is Schulam and Saria (2015), and we build off of their model for a univariate marker of disease trajectory that uses a GP with a highly structured mean function. Closest to our work in the application is Perotte

et al. (2015), who explore using time-series models to predict progression from CKD stage 3 to stage 4 in CKD patients. They use a standard Kalman filter to model multivariate laboratory data, which will not be as flexible as our GP-based models.

2.2.3 Joint Models for Longitudinal and Survival Data

There is a rich literature in biostatistics on joint models for longitudinal data and survival data, i.e. time-to-event data with right censoring. See Rizopoulos (2012) for a thorough introduction to these types of joint models. These models are constructed by specifying conditionally independent submodels for the longitudinal and time-to-event data; typically, some form of linear mixed effects model is used to model the longitudinal data, and a Cox regression is used for the survival data. The two submodels are then linked together in some fashion. The most common way this is accomplished is by using the unknown latent mean of the longitudinal model as a time-varying predictor in the Cox model. Other options also exist; Rizopoulos et al. (2014) consider different links between the submodels, and combine the resulting predictions using Bayesian model averaging. A slightly different class of joint models is presented in Proust-Lima et al. (2014). These models differ in that instead of the longitudinal value directly influencing the event rate, they consider latent subpopulations of individuals within which it is assumed there is a different average profile of both the longitudinal value and risk of the event. Most directly relevant to our joint modeling work in this chapter are methods for modeling longitudinal data and recurrent event data (Han et al., 2007; Liu and Huang, 2009; Kim et al., 2012; Musoro et al., 2015; Mbogning et al., 2015). Unlike traditional joint modeling in which an adverse event either never occurs or only occurs once, in this setting events may occur multiple times. This more accurately reflects the clinical setting in which our models will be used, since a patient who has a cardiovascular event is more likely to have additional events in the future.

However, all of these methods share several notable weaknesses. The specified parametric forms for their longitudinal models are simplistic, all being mixed effects models. Such models are inflexible and will fail to capture the types of trajectories that our proposed model can, through its mixture model and both long and short-term individual-specific deviations. In addition, these works as well as most of the literature on joint models rely on computationally expensive inference algorithms, thereby limiting their use to small datasets. Typically expectation-maximization (EM) or gradient-based methods are employed for Maximum Likelihood Estimation, or computationally expensive Markov Chain Monte Carlo (MCMC) in Bayesian settings. It is uncommon to find a published joint model applied to a dataset of more than a few thousand patients; a rare exception is Soleimani et al. (2017), which appeared after the initial development of the work in this chapter. The scalable variational inference algorithm developed in this chapter is much more efficient, facilitating use in large-scale clinical applications using EHRs, where there may be tens or hundreds of thousands of patients.

2.3 Proposed Joint Model

Our proposed hierarchical latent variable model jointly models longitudinal and point process data by creating different submodels for each type of data, with shared latent variables for each patient inducing dependencies between their two data types. Assume there are N patients, let $\mathbf{y}_i = \{y_{ij}\}_{j=1}^{N_i}$ denote the N_i observed readings of eGFR for patient i at times $\mathbf{t}_i = \{t_{ij}\}_{j=1}^{N_i}$, and let $\mathbf{u}_i = \{u_{ik}\}_{k=1}^{K_i}$ denote the K_i cardiac events patient i experiences (note that K_i may be 0). Let $\mathbf{x}_i$ denote a vector of covariates measured at baseline (e.g. age, gender, race, other comorbidities). Let T_i^- be the time patient i is first seen in our sample of their health record, and T_i^+ the final time they are observed. Let $z_i, \mathbf{b}_i, \mathbf{f}_i$ and v_i be a set of shared hierarchical

latent variables for each patient i, to be defined subsequently. Conditioned on these latent variables to be learned during inference, we make a common conditional independence assumption in joint modeling that the conditional likelihood for patient i factorizes (we implicitly condition on $\mathbf{x}_i$ throughout):

$$p(\mathbf{y}_i, \mathbf{u}_i | z_i, \mathbf{b}_i, \mathbf{f}_i, v_i,) = p(\mathbf{y}_i | z_i, \mathbf{b}_i, \mathbf{f}_i) p(\mathbf{u}_i | z_i, \mathbf{b}_i, \mathbf{f}_i, v_i). \tag{2.1}$$

2.3.1 Longitudinal Submodel

We use a recently proposed model for disease trajectories for our longitudinal submodel that was shown to be extremely flexible and accurate at modeling continuous functions of disease progression (Schulam and Saria, 2015). Given the set of latent variables for patient i, the longitudinal variables are conditionally independent, i.e. $p(\mathbf{y}|z_i, b_i, f_i) = \prod_{j=1}^{N_i} p(y_{ij}|z_i, \mathbf{b}_i, \mathbf{f}_i)$. The model assumes each observed longitudinal value is a normally distributed random variable containing a population component, a subpopulation component, an individual component, and a structured noise component:

$$y_i(t) = m_i(t) + \epsilon_i(t), \quad \epsilon_i(t) \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2) \tag{2.2}$$

$$m_i(t) = \mathbf{\Phi}_p(t)^\top \mathbf{\Lambda} \mathbf{x}_{ip} + \mathbf{\Phi}_z(t)^\top \boldsymbol{\beta}_{z_i} + \mathbf{\Phi}_l(t)^\top \mathbf{b}_i + f_i(t). \tag{2.3}$$

The first term in Equation 2.3 is the population component, where $\mathbf{\Phi}_p(t) \in \mathbb{R}^{d_p}$ is a fixed basis expansion of time, $\mathbf{\Lambda} \in \mathbb{R}^{d_p \times q_p}$ is a coefficient matrix, and $\mathbf{x}_{ip} \in \mathbb{R}^{q_p}$ is a vector of baseline covariates.

The second term in Equation 2.3 is the subpopulation component, where it is assumed person i belongs to latent subpopulation $z_i \in \{1, \ldots, G\}$. Each subpopulation is associated with a unique disease trajectory represented using B-splines, in particular, $\mathbf{\Phi}_z(t) \in \mathbb{R}^{d_z}$ is a fixed B-spline basis expansion of time with $\boldsymbol{\beta}_g \in \mathbb{R}^{d_z}$ the coefficient vector for group g, and $\boldsymbol{B} = \{\boldsymbol{\beta}_g\}_{g=1}^G$. We assign z_i a multinomial logistic

regression prior that depends on baseline covariates $\mathbf{x}_{iz} \in \mathbb{R}^{q_z}$:

$$p(z_i = g) = \frac{\exp\{\boldsymbol{w}_g^\top \mathbf{x}_{iz}\}}{\sum_{g'=1}^{G} \exp\{\boldsymbol{w}_{g'}^\top \mathbf{x}_{iz}\}}, \tag{2.4}$$

where $\boldsymbol{W} = \{\boldsymbol{w}_g\}_{g=1}^{G}$ are regression coefficients with $\boldsymbol{w}_1 \equiv \mathbf{0}$ for identifiability.

The third term in Equation 2.3 is the individual component, allowing for individual-specific long-term deviations in trajectory that are learned dynamically as more data is available. $\boldsymbol{\Phi}_l(t) \in \mathbb{R}^{d_l}$ is a fixed basis expansion of time, and $\mathbf{b}_i \in \mathbb{R}^{d_l}$ is a random effect for patient i, with prior $\mathbf{b}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_b)$.

Finally, $f_i(t)$ is the structured noise process that captures transient trends in disease trajectory. This is modeled using a zero-mean Gaussian process with Ornstein-Uhlenbeck covariance function

$$K_{OU}(t_1, t_2) = \sigma_f^2 \exp\{-\frac{|t_1 - t_2|}{l}\}. \tag{2.5}$$

This kernel is well-suited for this task, as it is mean-reverting and has no long-range dependence between deviations (Schulam and Saria, 2015). We could equivalently formulate this model as a GP with a highly structured mean function given by the first three terms in Equation 2.3.

2.3.2 Point Process Submodel

We choose to model the times $\mathbf{u}_i = \{u_{ik}\}_{k=1}^{K_i}$ that a person has an adverse event as a Poisson process. A common choice for the rate function from related literature in survival analysis corresponds to the hazard function from the Cox proportional hazards model. We make this choice in this work, for reasons both of simplicity and also computational efficiency as we discuss later. The conditional likelihood for the Poisson process for patient i on the interval $[T_i^-, T_i^+]$, with events at times $\{u_{ik}\}_{k=1}^{K_i}$,

is given by:

$$p(\mathbf{u}_i|z_i, \mathbf{b}_i, \mathbf{f}_i, v_i) = \prod_{k=1}^{K_i} r_i(u_{ik}) \exp\{-\int_{T_i^-}^{T_i^+} r_i(t)dt\}, \tag{2.6}$$

where we specify the rate function for patient i as:

$$r_i(t) = r_0(t) \exp\{\boldsymbol{\gamma}^\top \mathbf{x}_{ir} + \alpha m_i(t) + \delta m_i'(t) + v_i\}. \tag{2.7}$$

We assume that $r_0(t)$ is a piecewise constant function with jumps at fixed quantiles of the event times, and heights $\boldsymbol{a} = \{a_l\}_{l=1}^{N_r}$. The parameter $\boldsymbol{\gamma} \in \mathbb{R}^{q_r}$ specifies the association between baseline covariates $\mathbf{x}_{ir} \in \mathbb{R}^{q_r}$ and the risk for an event, while parameters α and δ specify the association between the risk for an event and the expected mean and expected slope of the longitudinal variable at that time, respectively. Note since $f_i(t)$ with an OU kernel is not differentiable, we let $m_i'(t) \equiv \frac{d}{dt} m_i(t)$ be the sum of the slopes of the first three terms in Equation 2.3. Finally, the latent variable v_i, with prior $v_i \sim \mathcal{N}(0, \sigma_v^2)$, represents an additional random effect (called a frailty term in survival analysis), multiplicatively adjusting an individual's overall risk for events. In order to compute the likelihood, we must compute the definite integral in Equation 2.6 numerically. We find that the trapezoid rule works fine, although other options such as Gaussian quadrature are also possible. The full set of model parameters to be learned are $\boldsymbol{\Theta} = \{\boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{B}, \boldsymbol{a}, \boldsymbol{\gamma}, \alpha, \delta\}$

2.4 Variational Inference for Joint Models

As with most complex probabilistic generative models, the computational problem associated with fitting the model is estimation of the posterior distribution of latent variables and model parameters given the observed data. Exact computation of the posterior is intractable, and requires approximation to compute. To this end, we develop a mean field variational inference algorithm to approximate the posterior distribution of interest (Jordan et al., 1999).

Variational methods transform the task of posterior inference into an optimization problem. The optimization problem posed by variational inference is to find a distribution q in some approximating family of distributions that is close in KL divergence to the true posterior. The choice of approximating family aims to balance tractability, allowing for efficient computation, with flexibility, allowing for expressive approximations to the true posterior. Equivalently, the problem can be viewed as maximizing what is known as the evidence lower bound (ELBO) $\mathcal{L}(q)$, which forms a lower bound on the marginal likelihood $p(\mathbf{y}, \mathbf{u})$ of our model (Bishop, 2006):

$$\log p(\mathbf{y}, \mathbf{u}) \geqslant \mathcal{L}(q) \tag{2.8}$$

$$\equiv \mathbb{E}_q[\log p(\mathbf{y}, \mathbf{u}, \mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta}) - \log q(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})] \tag{2.9}$$

$$= \mathbb{E}_q[\log q(\mathbf{y}, \mathbf{u}|\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})] - KL(q(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})||p(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})). \tag{2.10}$$

Equation 2.10 expresses the ELBO in terms of the expected log-likelihood of the data $\mathbb{E}_q[\log q(\mathbf{y}, \mathbf{u}|\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})]$, and the KL divergence between the prior $p(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})$ and the approximate posterior $q(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta})$. This variational objective mirrors the usual balance between likelihood and prior, as this first term encourages densities that place mass on settings of the latent variables and parameters that well explain the data, while the second term regularizes by encourages densities close to the prior.

2.4.1 Variational Approximation

Recall for our model that the model parameters are $\boldsymbol{\Theta} = \{\boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{B}, \boldsymbol{a}, \boldsymbol{\gamma}, \alpha, \delta\}$, and the local latent variables specific to each person are their subpopulation assignment z_i, random effects $\mathbf{b}_i$ and v_i, and structured noise function $\mathbf{f}_i$. The joint distribution for our model can be expressed as:

$$p(\mathbf{y}, \mathbf{u}, \mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \boldsymbol{\Theta}) = p(\boldsymbol{\Theta}) \prod_{i=1}^{N} p(\mathbf{y}_i|z_i, \mathbf{b}_i, \mathbf{f}_i, \boldsymbol{\Theta}) p(\mathbf{u}_i|z_i, \mathbf{b}_i, \mathbf{f}_i, v_i, \boldsymbol{\Theta}) p(z_i|\boldsymbol{\Theta}) p(\mathbf{b}_i) p(\mathbf{f}_i) p(v_i) \tag{2.11}$$

We make the mean field assumption for the variational distribution, which assumes that in the approximate posterior q, all the latent variables are independent. This implies that $q(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{v}, \mathbf{\Theta}) = q(\mathbf{\Theta}) \prod_{i=1}^{N} q_i(z_i, \mathbf{b}_i, \mathbf{f}_i, v_i)$, where:

$$q_i(z_i, \mathbf{b}_i, \mathbf{f}_i, v_i) = q_i(z_i|\boldsymbol{\nu}_{z_i}) q_i(\mathbf{b}_i|\boldsymbol{\mu}_{b_i}, \mathbf{\Sigma}_{b_i}) q_i(v_i|\mu_{v_i}, \sigma^2_{v_i}) q_i(\mathbf{f}_i). \tag{2.12}$$

The assumed variational distributions for z_i, $\mathbf{b}_i$, and v_i are the same family as their prior distribution, i.e. multinomial, multivariate normal, and univariate normal. For the variational form for $\mathbf{f}_i$, we adapt ideas from the variational learning for sparse GPs literature to approximate the true posterior over $\mathbf{f}_i$ (Lloyd et al., 2015; Titsias, 2009). In order to evaluate the ELBO in Equation 2.10, we will need to evaluate $\mathbb{E}_{q_i}[\mathbf{f}_i]$ at times $\mathbf{t}_i$ for the longitudinal likelihood, as well as at $\mathbf{u}_i$ and at a grid of times $\mathbf{t}_i^{\text{grid}}$ for the point process likelihood (for the numerical integration). We choose to treat the observed observation times $\mathbf{t}_i$ as pseudo-inputs; this helps reduce overfitting and reduces the number of variational parameters to learn. In particular, we assume:

$$q_i(f_i(\mathbf{t}_i), f_i(\mathbf{u}_i), f_i(\mathbf{t}_i^{\text{grid}})) = p(f_i(\mathbf{u}_i), f_i(\mathbf{t}_i^{\text{grid}})|f_i(\mathbf{t}_i)) q(f_i(\mathbf{t}_i)|\boldsymbol{\mu}_{f_i}, \mathbf{\Sigma}_{f_i}). \tag{2.13}$$

We allow a free-form multivariate Gaussian distribution for $\mathbf{f}_i$ at the longitudinal observation times $\mathbf{t}_i$, and use a so-called conditional Gaussian process for the distribution at $\mathbf{u}_i, \mathbf{t}_i^{\text{grid}}$, i.e. the true conditional distribution of the joint multivariate normal:

$$\mathbf{f}_i|f_i(\mathbf{t}_i) \sim \mathcal{GP}(\mu(t), \Sigma(t, t')) \tag{2.14}$$

$$\mu(t) = \mathbf{k}_{t,t_i} \mathbf{K}^{-1}_{\mathbf{t}_i,\mathbf{t}_i} f_i(\mathbf{t}_i) \tag{2.15}$$

$$\Sigma(t, t') = K(t, t') - \mathbf{k}_{t,t_i} \mathbf{K}^{-1}_{t_i,t_i} \mathbf{k}_{t_i,t'} \tag{2.16}$$

where $\mathbf{K}_{t_i,\mathbf{t}_i}, \mathbf{k}_{t,\mathbf{t}_i}, K(t, t')$ are matrices/vectors/scalars of covariances between relevant times, using the OU covariance kernel K defined in Equation 2.5.

Although priors on the model parameters $\mathbf{\Theta}$ may be imposed, i.e. log-normal on $\mathbf{a}$ and normal on the rest, in our work we learn their maximum likelihood estimate

(MLE) instead, and let $q(\boldsymbol{\Theta})$ be a delta function. Thus, the goal of our variational algorithm is to learn optimal variational parameters $\boldsymbol{\lambda}_i = \{\boldsymbol{\nu}_{z_i}, \boldsymbol{\mu}_{b_i}, \boldsymbol{\Sigma}_{b_i}, \mu_{v_i}, \sigma^2_{v_i}, \boldsymbol{\mu}_{f_i}, \boldsymbol{\Sigma}_{f_i}\}$ for each individual i, as well as a point estimate $\hat{\boldsymbol{\Theta}}$ for the model parameters. In practice, we optimize the Cholesky decompositions $\boldsymbol{L}_{b_i}, \boldsymbol{L}_{f_i}$ for the covariance matrices $\boldsymbol{\Sigma}_{b_i}, \boldsymbol{\Sigma}_{f_i}$.

2.4.2 Evidence Lower Bound

We briefly derive the ELBO for our joint model, which has an exact analytical form (up to a numerical integration approximation for the integral in Equation 2.6). We can rewrite the expression for the ELBO in Equation 2.10 as:

$$\mathcal{L}(q) = \sum_{i=1}^{N} \mathcal{L}(q_i), \tag{2.17}$$

$$\begin{aligned}\mathcal{L}(q_i) &= \mathbb{E}_{q_i}[\log\, p(\mathbf{y}_i|z_i, \mathbf{b}_i, \mathbf{f}_i, \boldsymbol{\Theta}) + \log\, p(\mathbf{u}_i|z_i, \mathbf{b}_i, \mathbf{f}_i, v_i, \boldsymbol{\Theta})] \\ &\quad - KL(q_i(\mathbf{b}_i)||p(\mathbf{b}_i)) - KL(q_i(v_i)||p(v_i)) \\ &\quad - KL(q_i(z_i)||p(z_i)) - KL(q_i(f_i(\mathbf{t}_i))||p(f_i(\mathbf{t}_i))).\end{aligned} \tag{2.18}$$

Computation of the KL divergence terms are standard. We focus our attention on the expected log-likelihood, i.e. the first two terms.

The first term in Equation 2.18 is the variational expectation of the log-likelihood for the longitudinal submodel. To compute this, we need to calculate the variatonal expectation $\mathbb{E}_{q_i}[(\mathbf{y}_i - m_i(\mathbf{t}_i))^\top(\mathbf{y}_i - m_i(\mathbf{t}_i))]$ with respect to $q_i(\mathbf{b}_i)$, $q_i(z_i)$, and $q_i(\mathbf{f}_i)$; this is straightforward using standard results on the expectation of a quadratic form.

The second term in Equation 2.18 is the variational expectation of the log-likelihood for the point process submodel. We can easily compute $\mathbb{E}_{q_i}[\log\, r_i(u_{ik})]$ for each term in the summation in Equation 2.6. In order to compute $\mathbb{E}_{q_i}[m_i'(u_{ik})]$ we simply use the time derivatives of the bases $\boldsymbol{\Phi}_p', \boldsymbol{\Phi}_z', \boldsymbol{\Phi}_l'$. The only nontrivial term is $\mathbb{E}_{q_i}[f_i(u_{ik})]$. However, due to conjugacy, we have an exact variational distribution

for $f_i(t)$, for arbitrary t:

$$q_i(f_i(t)) = \int p(f_i(t)|f_i(\mathbf{t}_i))q_i(f_i(\mathbf{t}_i)) \equiv \mathcal{GP}(f_i; \mu(t), \Sigma(t,t')) \tag{2.19}$$

$$\mu(t) = \mathbf{k}_{t,\mathbf{t}_i}\mathbf{K}^{-1}_{\mathbf{t}_i,\mathbf{t}_i}\boldsymbol{\mu}_{f_i},\ \Sigma(t,t') = K(t,t') - \mathbf{k}_{t,\mathbf{t}_i}\mathbf{K}^{-1}_{\mathbf{t}_i,\mathbf{t}_i}\mathbf{k}_{\mathbf{t}_i,t'} + \mathbf{k}_{t,\mathbf{t}_i}\mathbf{K}^{-1}_{\mathbf{t}_i,\mathbf{t}_i}\boldsymbol{\Sigma}_{f_i}\mathbf{K}^{-1}_{\mathbf{t}_i,\mathbf{t}_i}\mathbf{k}_{\mathbf{t}_i,t'}. \tag{2.20}$$

The last term to compute is the variational expectation of the integral in Equation 2.6. Since we approximate it numerically, we need to evaluate $\mathbb{E}_{q_i}[r_i(t)]$ for arbitrary times t:

$$\mathbb{E}_{q_i}[r_i(t)] = r_0(t)e^{\boldsymbol{\gamma}^\top \mathbf{x}_{ir}}\mathbb{E}_{q_i}[e^{\alpha m_i(t) + \delta m_i'(t) + v_i}]. \tag{2.21}$$

Using the mean field independence assumption, the above expectation of products factorizes. Some of the terms in the product involve taking expectations of Gaussian distributions; for instance, $q_i(v_i) \sim \mathcal{N}(\mu_{v_i}, \sigma^2_{v_i})$, so e^{v_i} is distributed according to a log-normal, and thus $\mathbb{E}_{q_i}[e^{v_i}] = e^{\mu_{v_i} + \sigma^2_{v_i}/2}$. Likewise, expectations involving exponentials of $\mathbf{b}_i$ and $\mathbf{f}_i$ involve multivariate log-normal distributions, since their variational distributions are multivariate normals. These expectations have closed forms, since $\mathbb{E}[\boldsymbol{y}] = e^{\boldsymbol{\mu} + \text{diag}(\boldsymbol{\Sigma})/2}$, where $\boldsymbol{y} = e^{\boldsymbol{x}}$ is multivariate log-normal distributed, with $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus, the ELBO has an exact analytic form, which will make it easy to use automatic differentiation to optimize it using gradient-based methods.

2.4.3 Solving the Optimization Problem

In traditional settings for variational inference, the objective function is iteratively optimized by maximizing the variational parameters associated with each latent variable or parameter, holding the rest fixed. In models where the log complete conditional distributions (log of the conditional distribution of each latent variable given everything else) have analytic expectations with respect to the variational approximation, closed form EM-style updates are available for the variational parameters. This convenient property is typically observed in conditionally conjugate models,

where each log complete conditional will be in the exponential family (Ghahramani and Beal, 2001).

Recently there have been many approaches to apply variational methods to complex non-conjugate models. Usually, it is intractable to even evaluate the ELBO analytically, since either the expected log-likelihood term or the KL divergence term (or both) in Equation 2.10 do not have a closed form. In these cases, variational algorithms have been developed that rely on sampling from the variational approximation (Ranganath et al., 2014; Rezende et al., 2014). However, because of the particular form we chose for r_i, it is possible to calculate a closed form approximation to the ELBO for our model, as we previously showed. As such, we can simply apply an automatic differentiation package *autograd* [1] in Python to compute analytic gradients in order to optimize the bound. At each iteration of the algorithm, we optimize the local variational parameters in parallel using exact gradients. To optimize the global parameters, we turn to stochastic optimization.

Stochastic optimization has become a commonly used tool in variational inference. Rather than using every single observation to compute the gradient of the ELBO with respect to $\mathbf{\Theta}$, we can compute a noisy gradient based on a sampled batch of observations (Hoffman et al., 2013). As long as the noisy gradient is unbiased and the learning rate ρ_t at each iteration satisfies the Robbins-Monro conditions ($\sum_{t=1}^{\infty} \rho_t = \infty,\ \sum_{t=1}^{\infty} \rho_t^2 < \infty$), the stochastic optimization procedure will converge to a local maximum (Robbins and Monro, 1951). To set the learning rate we use the AdaGrad algorithm, which adaptively allows for a different learning rate for each parameter. The learning rate for each parameter is scaled by the square root of a running sum of the squares of historical gradients (Duchi et al., 2011).

To compute noisy gradients of the ELBO with respect to $\mathbf{\Theta}$, we randomly sample S observations $\{\mathbf{y}_s, \mathbf{u}_s\}_{s=1}^{S}$ at each iteration, and compute the gradient of the rescaled

[1] `https://github.com/HIPS/autograd`

$\hat{\mathcal{L}}(q) \equiv \frac{N}{S}\sum_{s=1}^{S}\mathcal{L}(q_s)$, which equals $\mathcal{L}(q)$ in expectation. Algorithm 1 summarizes the procedure to learn an approximate posterior for the local latent variables and a point estimate for the model parameters.

Data: data $\mathbf{y}, \mathbf{u}$; hyperparameters.
Result: point estimate $\hat{\boldsymbol{\Theta}}$, approximate posteriors q_i.
Initialize global parameters $\boldsymbol{\Theta}$.
repeat
  Randomly sample data for S patients, $\{\mathbf{y}_s, \mathbf{u}_s\}_{s=1}^{S}$. **for** *s = 1:S in parallel* **do**
    Optimize local variational parameters for q_s via gradient ascent.
  end
  Compute the noisy gradient for $\boldsymbol{\Theta}$.
  Update $\boldsymbol{\Theta}$ using AdaGrad.
until *convergence of the ELBO*;

Algorithm 1: Overview of stochastic variational inference algorithm for the proposed joint model.

2.5 Joint Model Empirical Study

In this section we describe our experimental setup and results on our real dataset.

2.5.1 Chronic Kidney Disease Dataset

Our dataset comprises longitudinal and cardiac event data from 23,450 patients with stage 3 CKD or higher within the Duke University health system. IRB approval (#Pro00066690) was obtained for this work. We first created an initial cohort of roughly 600,000 patients that had at least one encounter in the health system in the year prior to Feb. 1, 2015. This includes all types of encounters within the health system, including inpatient, outpatient, and emergency department visits over a span of roughly 20 years. From this, we filtered to patients who had at least ten recorded values for serum creatinine, the laboratory value required to calculate eGFR. We next filtered to patients that had Stage 3 CKD or higher, indicative of moderate to severe kidney damage, defined as two eGFR measurements less than 60 mL/min separated by at least 90 days. Finally, since the recorded eGFR values are extremely noisy and

eGFR is only a valid estimate of kidney function at steady state, we take the mean of eGFR readings in monthly time bins for each patient. Rapid fluctuations in acute illness are related to long term risk, but we have not yet explicitly incorporated this into our modeling.

The adverse events of interest in our experiments are AMIs and CVAs, and these were identified using ICD-9 codes, with the Clinical Classifications Software[2] from the Agency for Healthcare Research and Quality. After identifying codes for CVAs and AMIs we aggregate them using the mean date among all relevant codes within monthly bins, to account for multiple codes in a short time period that refer to the same clinical event. There are numerous ongoing efforts to develop improved algorithms to identify chronic medical conditions and incident clinical events using a wide assortment of clinical data. Our model is agnostic to the particular algorithm used to identify clinical events.

After cleaning and transforming the raw EHR data, we obtained a longitudinal set of eGFRs for each patient, and dates of CVA and AMI diagnoses. On average each patient has 22.9 eGFR readings (std dev 13.6; median 19.0). In order to align the patients on a common time axis, for each patient we fix $t = 0$ to be their first recorded eGFR reading below 60 mL/min. 13.4% of patients had at least one code for AMI (among those with at least one: mean 4.1, std dev 7.1, median 2.0), and likewise 17.4% of patients had at least one code for CVA (mean 6.4, std dev 13.3, median 3.0). We use the same set of baseline covariates for $\mathbf{x}_{ip}, \mathbf{x}_{iz}, \mathbf{x}_{ir}$: baseline age, race and gender, and indicator variables for hypertension and diabetes. Note that $\mathbf{x}_{ip}, \mathbf{x}_{iz}$ include an intercept while $\mathbf{x}_{ir}$ does not.

For the experiments, we used ten fold cross validation with training sets of 21,105 patient records and test sets of 2,345 records. We fit separate joint models for CVA events and AMI events.

[2] `http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp`

2.5.2 Evaluation Metrics

After learning a point estimate for the global model parameters during training, they are held fixed. Then, an approximate posterior is fit to each patient in the test set, where we allow the learning algorithm to see the first 60% of a patient's eGFR trajectory (and any events before then) and hold out the remaining 40% (and future events). Predictions about future disease trajectory and adverse events are made by drawing samples from the approximate posterior predictive distribution.

We evaluate our model on two tasks to asses predictive performance of each submodel. For the longitudinal submodel, we compute the mean squared error (MSE) and mean absolute error (MAE) for predictions about held-out eGFR values. For the point process submodel, we view the problem of predicting whether any event will occur in a given future time window (in our experiments, 1-5 years) as a binary classification problem. We report the area under the ROC curve (AUROC) and area under the precision-recall curve (AUPR) as evaluation metrics for each binary classification task. Calculating the probability of an event in a future time window $[T_i, T_i + c]$ for person i is easily computed as $1 - \exp\{-\int_{T_i}^{T_i+c} r_i(t)dt\}$.

2.5.3 Baselines

For the longitudinal submodel, we compare against the model in Schulam and Saria (2015), since we use their model as our longitudinal submodel. However, because our model was trained jointly with the point process submodel we do not in general learn the same model parameters, since the parameters for the learned trajectories are also influenced by the event data.

For the point process submodel, we compare against two standard baselines. The first is a simple Cox proportional hazards model from survival analysis, where we use the same set of time independent covariates x_{ir} as in our model. The likelihood is the same as Equation 2.6, but now $r_i(t) = r_0(t) \exp\{\boldsymbol{\gamma}^\top \mathbf{x}_{ir}\}$. We also compare against a

Cox model with time-dependent covariates, where $r_i(t) = r_0(t)\ \exp\{\boldsymbol{\gamma}^\top \mathbf{x}_{ir} + \alpha y_i(t)\}$, with $y_i(t)$ a step function denoting the most recent observed eGFR up until time t. Due to the lack of scalable inference algorithms for related works from the joint modeling literature, we were unable to compare against them on our large patient cohort.

2.5.4 Hyperparameters

We learn point estimates for hyperparameters $\sigma_\epsilon, \boldsymbol{\Sigma}_b, \sigma_v, \sigma_f, l$ by maximizing the ELBO with respect to them. Additional hyperparameters include G, N_r, and the choice of basis expansions $\boldsymbol{\Phi}_p, \boldsymbol{\Phi}_z, \boldsymbol{\Phi}_l$ in the longitudinal submodels. We let $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_l$ be linear basis functions of time, thus allowing for population covariates and individual heterogeneity to shift the intercept and slope of eGFR trajectory. We let $\boldsymbol{\Phi}_z$ be a B-spline expansion of time with degree two and twelve knots at equally spaced quantiles of eGFR observation times. We fix $G = 15$ and $N_r = 9$. Finally, we set the global scale parameter for AdaGrad to 0.1, and subsample 250 observations at a time. We experimented with other values for these fixed hyperparameters without major changes in performance.

2.5.5 Results

Figure 2.2 highlights the results from the longitudinal submodel, where we present the mean MSEs and MAEs across the test sets. The longitudinal submodel from our joint model performs slightly better than the method of Schulam and Saria (2015) fit independently to the eGFR values. Figure 2.3 highlight the results from the point process submodel. Our proposed joint model performs substantially better than the two baselines at predicting future events, in terms of both AUROC and AUPR.

In addition, in this dataset it appears that prediction of CVA events is slightly easier than prediction of AMIs. For the CVA joint model, we estimate that $\alpha =$

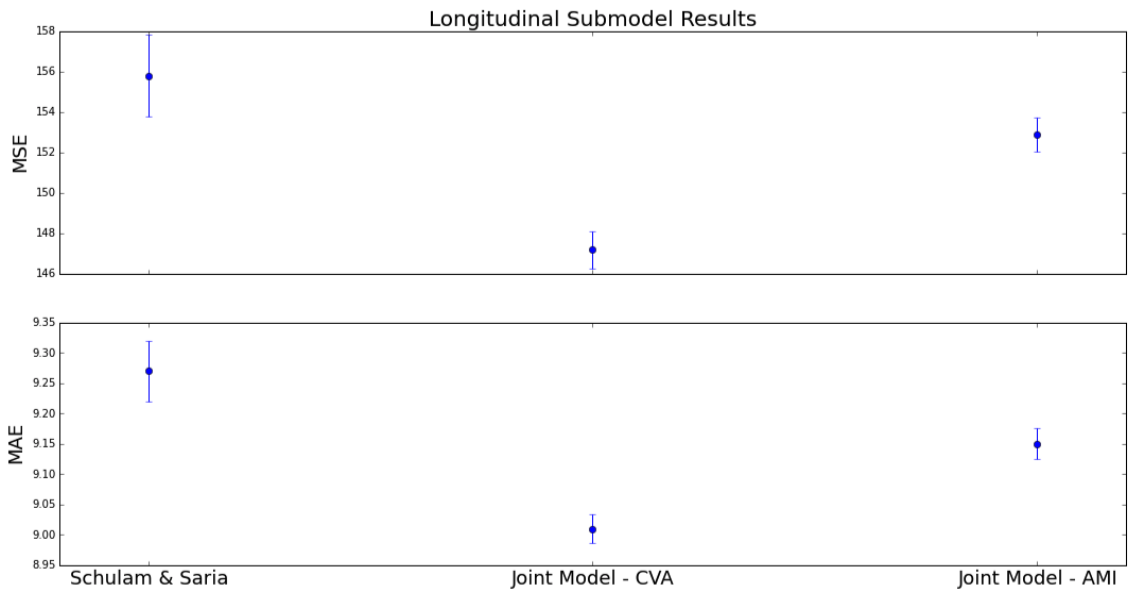

FIGURE 2.2: Mean MSE and MAE from longitudinal submodels. Error bars are one standard error.

-0.063 and $\delta = -0.061$ while for the AMI joint model, $\alpha = -0.158$ and $\delta = -0.069$ (standard errors for all four estimates < 0.01, from the cross validation). The signs of these parameters agree with clinical intuition that patients with lower overall eGFR values and more rapid eGFR declines should be at higher risk for adverse events. It appears there is a slightly stronger association between eGFR trajectory and risk for AMIs compared to CVAs.

Figure 2.4 shows an example of dynamic predictions over time for a test patient. In the three rows of the figure, we make predictions about the test patient after observing the first 25%, 50% and 75% of their disease trajectory and adverse events (in this example, CVAs). For each row we relearn the patient's parameters using information to the left of the vertical light blue line. As we observe more data, the longitudinal model updates its prediction about future disease trajectory and provides a reasonable forecast for the steady decline of this patient's eGFR. In the second row, as the model sees that the patient's trajectory is decreasing faster than in the first row, it correspondingly increases the probability of a future event. In the

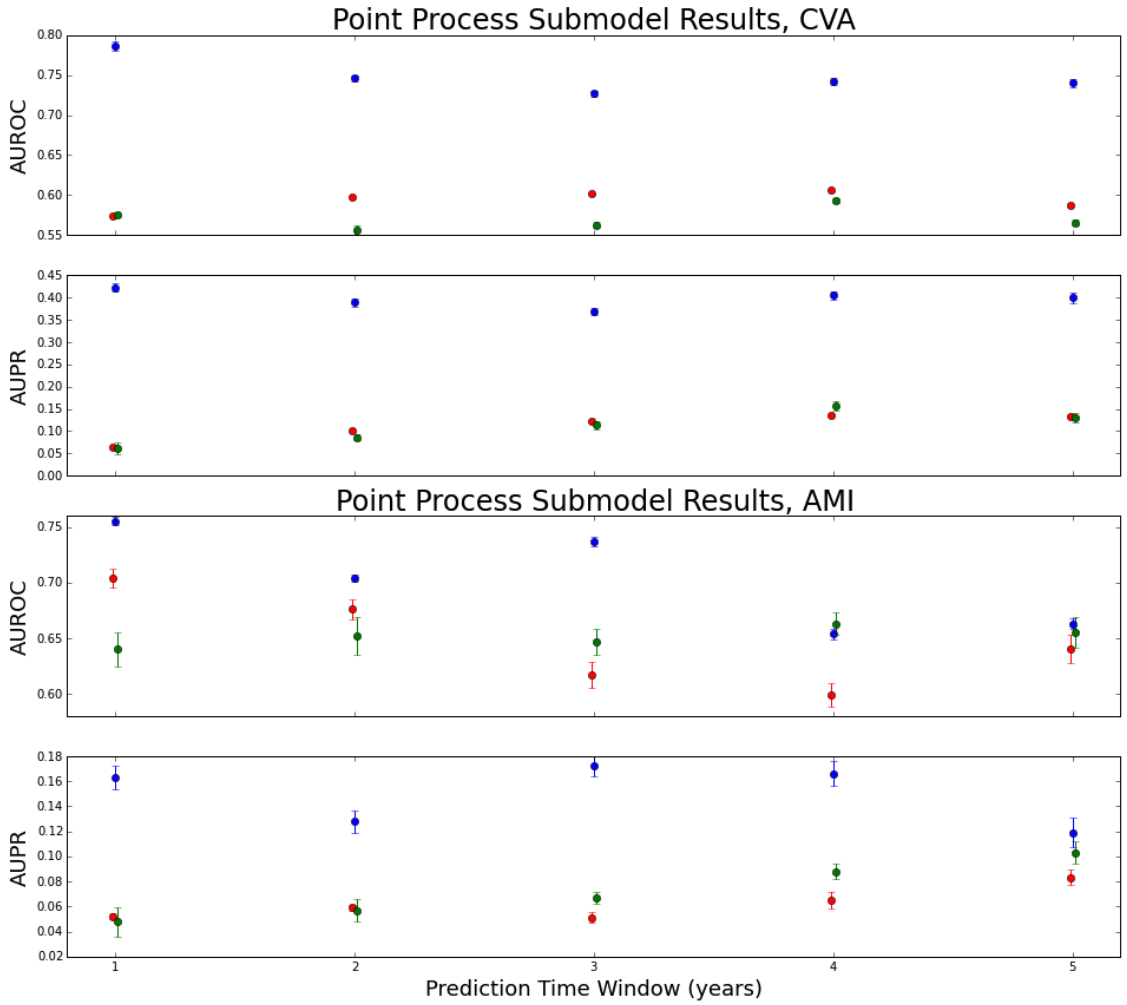

FIGURE 2.3: Mean AUROC and AUPR for CVA and AMI events. Blue is proposed Joint Model, red is Cox, green is time-varying Cox. Error bars are one standard error.

third row, after the model sees the patient's first CVA event, it further increases the probability of a future event.

2.6 Proposed Multivariate Disease Trajectory Model

We now transition to the second model considered in this chapter. Our interest now is in using the trajectories of other relevant labs to improve our predictions about the future trajectory of a target clinical marker, in our case eGFR. Predictions about

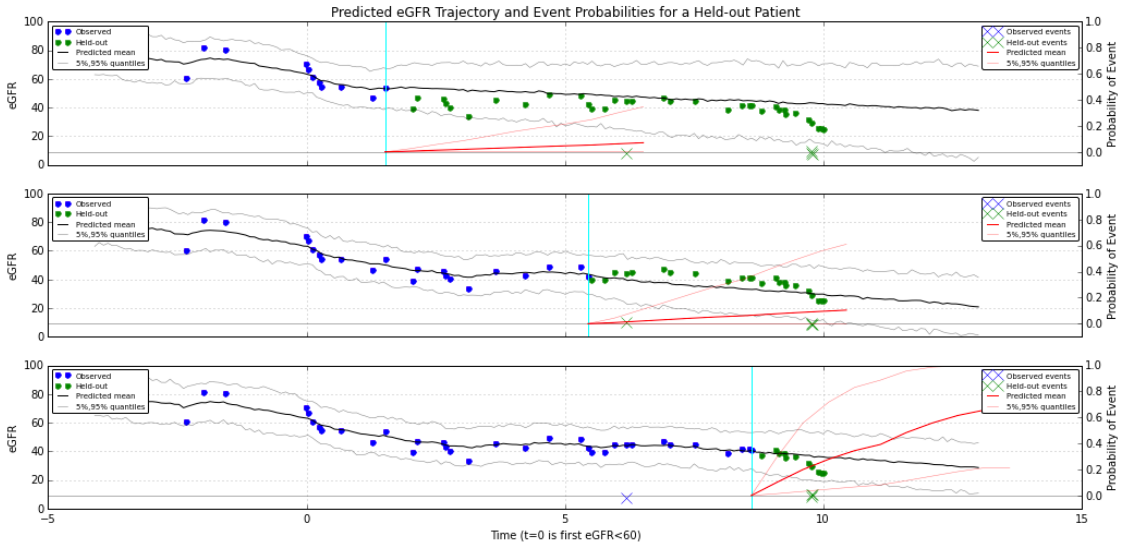

FIGURE 2.4: Dynamic predictions from our joint model. In each row, the parameters for this individual are refit as more data is made available (information to the left of the light blue lines is used to refit parameters). Blue circles and x's correspond to observed eGFR readings and CVA events, while green correspond to yet-unseen data.

the future trajectories of the other clinical variables can also be useful on their own. Figure 2.5 shows additional clinical data from the same patient in the beginning of this chapter, in Figure 2.1. Although predicting decline in future eGFR is our ultimate goal, information contained in the other five lab variables help paint a better overall clinical portrait of this patient.

Our proposed hierarchical latent variable model jointly models each patient's multivariate longitudinal data by using a GP for each individual variable, with shared latent variables inducing dependence between the mean functions. In the univariate setting, our model reduces to the method in Schulam and Saria (2015) that we used as our longitudinal submodel in the joint model presented earlier in this chapter.

The presentation of our model for multivariate clinical markers is similar to that of the longitudinal submodel in Section 2.3.1. Let $\mathbf{y}_i(t) = (y_{i1}(t), \ldots, y_{iM}(t))^\top \in \mathbb{R}^M$ denote the $M-$dimensional trajectory of measurements obtained for individual i, let

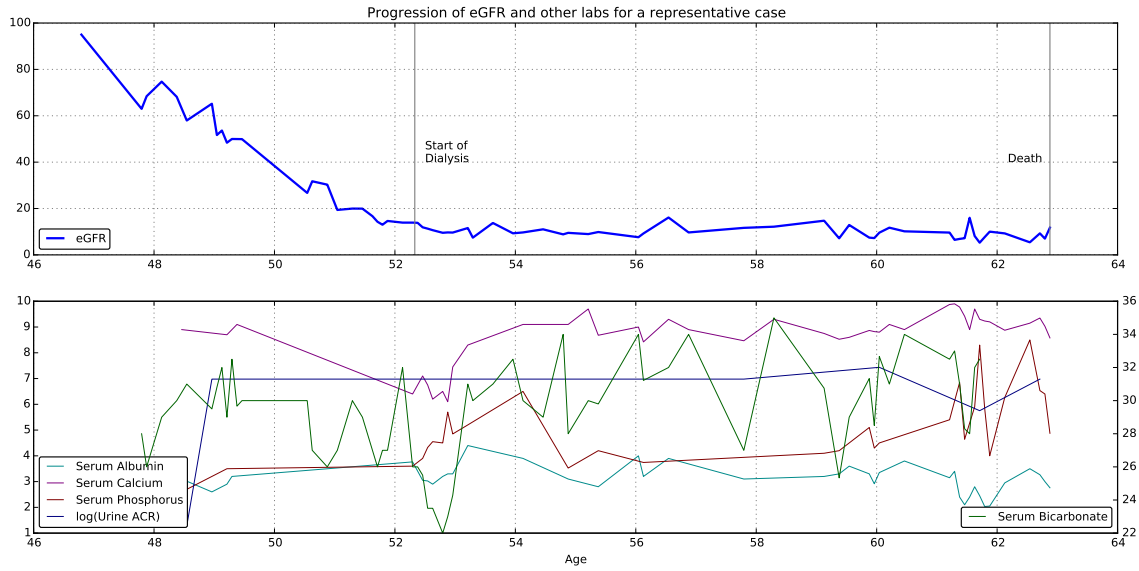

FIGURE 2.5: Clinical course of a patient who experienced a rapid progression of CKD (the same patient from Figure 2.1). Top plot shows estimated glomerular filtration rate (eGFR), an estimate of kidney function (60-100 is normal, <60 indicates clinically significant kidney disease). The bottom plot shows the trajectory of five other clinical labs relevant to kidney disease.

$\mathbf{y}_{im} = \{y_{im}(t_{imj})\}_{j=1}^{n_{im}}$ be the n_{im} observations for variable m at times t_{imj} for this individual, and let $\mathbf{y}_i = \{\mathbf{y}_{im}\}_{m=1}^{M}$. Let c_i, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iM})$, $\mathbf{b}_i = (\mathbf{b}_{i1}, \ldots, \mathbf{b}_{iM})$, and $\mathbf{f}_i = (f_{i1}, \ldots, f_{iM})$ be latent variables specific to individual i, to be defined shortly, and let $\mathbf{x}_i \in \mathbb{R}^q$ denote baseline covariates. We will make a similar conditional independence assumption for each person's longitudinal variables that we did for the joint model, and assume they factorize into submodels:

$$p(\mathbf{y}_i | c_i, \mathbf{z}_i, \mathbf{b}_i, \mathbf{f}_i) = \prod_{m=1}^{M} p(\mathbf{y}_{im} | c_i, z_{im}, \mathbf{b}_{im}, f_{im}). \tag{2.22}$$

For each longitudinal variable, we propose the following generative model:

$$y_{im}(t) \sim \mathcal{N}(\mu_{im}(t), \sigma_m^2) \tag{2.23}$$

$$\mu_{im}(t) = \mathbf{\Phi}_p^{(m)}(t)^\top \mathbf{\Lambda}^{(m)} \mathbf{x}_i + \mathbf{\Phi}_z^{(m)}(t)^\top \boldsymbol{\beta}_{z_{im}}^{(m)} + \mathbf{\Phi}_l^{(m)}(t)^\top \mathbf{b}_{im} + f_{im}(t) \tag{2.24}$$

$$f_{im}(t) \sim \mathcal{GP}(0, K_m), \quad K_m(t, t') = a_m^2 \exp\{-l_m^{-1}|t - t'|\} \tag{2.25}$$

$$z_{im}|c_i \sim \text{Multinomial}(\mathbf{\Psi}_{c_i}^{(m)}) \tag{2.26}$$

with priors on latent variables shared across all M submodels:

$$c_i \sim \text{Multinomial}(\boldsymbol{\pi}_i), \ \pi_{ig} = \frac{e^{\boldsymbol{w}_g^\top \mathbf{x}_i}}{\sum_{g'=1}^G e^{\boldsymbol{w}_{g'}^\top \mathbf{x}_i}} \tag{2.27}$$

$$\mathbf{b}_i = (\mathbf{b}_{i1}, \ldots, \mathbf{b}_{iM})^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_b). \tag{2.28}$$

Overall, this model is extremely similar to the longitudinal submodel in Equations 2.2 and 2.3. Each variable has its own submodel of this form, with information being shared across models through joint priors.

The first term in the mean in Equation 2.24 again admits population-level fixed effects for each lab using observed baseline covariates $\mathbf{x}_i \in \mathbb{R}^q$, e.g. gender, race, age (assume we use the same set of baseline covariates for each variable m). Analogous to before, $\mathbf{\Lambda}^{(m)} \in \mathbb{R}^{d_{p_m} \times q}$ is a coefficient matrix with $\mathbf{\Phi}_p^{(m)}(t) \in \mathbb{R}^{d_{p_m}}$ a basis expansion of time; in practice we use the same linear expansion of time with $d_{p_m} = 2$ so this term allows for a fixed intercept and slope for all labs.

Again as previously, the second term in Equation 2.24 is a subpopulation component, where it is assumed individual i's trajectory for variable m belongs to a latent subpopulation, denoted $z_{im} \in \{1, \ldots, G_m\}$. Each subpopulation is associated with a unique trajectory; in particular, $\mathbf{\Phi}_z^{(m)}(t) \in \mathbb{R}^{d_{z_m}}$ is a fixed B-spline basis expansion of time (for simplicity, assumed to be the same for each variable: degree two, with the same eight interior knots evenly spaced in time) with $\boldsymbol{\beta}_g^{(m)} \in \mathbb{R}^{d_{z_m}}$ the coefficient vector for subpopulation g and variable m.

The prior for each z_{im} in Equation 2.26 depends on the individual's "global" cluster $c_i \in \{1, \ldots, G\}$. Drawing on ideas from latent structure analysis for multivariate categorical data (Lazarsfeld and Henry, 1968), the c_i induce dependence among the M trajectory-specific clusters z_{im} when marginalized, as they have conditionally independent multinomial priors. Each of the G columns $\mathbf{\Psi}_g^{(m)}$ in the matrix $\mathbf{\Psi}^{(m)} \in \mathbb{R}^{G_m \times G}$ defines a distribution over the G_m values that z_{im} can take. The c_i then has a multinomial logistic regression prior in Equation 2.27 that depends on the baseline covariates $\mathbf{x}_i$, where $\{\boldsymbol{w}_g\}_{g=1}^G$ are regression coefficients with $\boldsymbol{w}_1 \equiv \mathbf{0}$ for identifiability. It is possible to construct an even more complex model where the priors for z_{im} also depend directly on covariates $\mathbf{x}_i$ as well as on $\mathbf{\Psi}^{(m)}$, but this greatly increases the number of parameters and did not appear to improve performance.

The third term in Equation 2.24 is again a random effects component, allowing for individual-specific long-term deviations in trajectory that are learned dynamically as more data becomes available. In practice $\mathbf{\Phi}_l^{(m)}(t) \in \mathbb{R}^{d_{l_m}}$ is a linear expansion of time with all $d_{l_m} = 2$, so that $b_{im} \in \mathbb{R}^{d_{l_m}}$ is a random slope and intercept vector for patient i. The only difference with the analogous random effect term in the univariate version of the model is that the overall vector $\mathbf{b}_i$ has a multivariate normal prior distribution in Equation 2.28, making the random effects dependent across labs.

Finally, $K_m(t, t')$ in Equation 2.25 is again the OU covariance function for a GP modeling short-term fluctuations for each variable, with parameters a_m, l_m. Note that as previously we can alternatively express our model as a GP with highly structured mean function, $\mu_{im}(t) \sim \mathcal{GP}(\mathbf{\Phi}_p^{(m)}(t)^\top \mathbf{\Lambda}^{(m)} \mathbf{x}_i + \mathbf{\Phi}_z^{(m)}(t)^\top \boldsymbol{\beta}_{z_{im}}^{(m)} + \mathbf{\Phi}_l^{(m)}(t)^\top \mathbf{b}_{im}, K_m)$. However, it will again be more convenient to explicitly represent the short-term deviations $\mathbf{f}_{im}$ from the GP in order to learn them directly during inference.

2.6.1 Variational Inference

As for the joint model presented earlier in the chapter, we will fit our multivariate GP model using stochastic variational inference. The setup is incredibly similar to the presentation in Section 2.4, so we leave out most of the details.

The model parameters to be learned are $\boldsymbol{\Theta} = \big\{\{\boldsymbol{\Lambda}^{(m)}, \boldsymbol{B}^{(m)}, \boldsymbol{\Psi}^{(m)}, a_p, l_p, \sigma_p^2\}_{p=1}^P, \boldsymbol{W}, \boldsymbol{\Sigma}_b\big\}$, and the local latent variables specific to each person are their global cluster assignment c_i, subpopulation assignments z_{ip}, random effects $\mathbf{b}_i$ and structured noise functions $\mathbf{f}_{ip}$. The joint distribution for our model can be expressed as:

$$p(\mathbf{y}, \mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{c}, \boldsymbol{\Theta}) = p(\boldsymbol{\Theta}) \prod_{i=1}^N p(\mathbf{b}_i|\boldsymbol{\Theta}) p(c_i|\boldsymbol{\Theta}) \prod_{m=1}^M p(\mathbf{y}_{im}|z_{im}, \mathbf{b}_{im}, \mathbf{f}_{im}, \boldsymbol{\Theta}) p(z_{im}|c_i, \boldsymbol{\Theta}) p(\mathbf{f}_{im}|\boldsymbol{\Theta}) \tag{2.29}$$

We again make a mean field assumption for the variational distribution, so all latent variables are independent in the approximate posterior. This implies $q(\mathbf{z}, \mathbf{b}, \mathbf{f}, \mathbf{c}, \boldsymbol{\Theta}) = q(\boldsymbol{\Theta}) \prod_{i=1}^N q_i(c_i, \mathbf{z}_i, \mathbf{b}_i, \mathbf{f}_i)$, where:

$$q_i(c_i, \mathbf{z}_i, \mathbf{b}_i, \mathbf{f}_i) = q_i(c_i|\boldsymbol{\nu}_{c_i}) q_i(\mathbf{b}_i|\boldsymbol{\mu}_{b_i}, \boldsymbol{\Sigma}_{b_i}) \prod_{m=1}^M q_i(z_{im}|\boldsymbol{\nu}_{z_{im}}) q_i(\mathbf{f}_{im}|\boldsymbol{\mu}_{f_{im}}, \boldsymbol{\Sigma}_{f_{im}}). \tag{2.30}$$

The assumed variational distributions for each latent variable are in the same family as their prior distribution, i.e. c_i and z_{im} are multinomials, and $\mathbf{b}_i$ is multivariate normal. For $\mathbf{f}_{im}$ we use a multivariate normal evaluated at all times at which variable m is observed for patient i. To evaluate $\mathbf{f}_{im}$ at additional times e.g. during model evaluation we use the conditional GP framework and treat the observed values as pseudo-inputs, from the sparse GP literature (Titsias, 2009), analogous to how $\mathbf{f}_i$ was treated for the joint model in Section 2.4. Finally, for all model parameters $\boldsymbol{\Theta}$ we learn a point estimate, so their variational distributions are delta functions. We impose vague normal priors on $\boldsymbol{\Lambda}^{(m)}, \boldsymbol{B}^{(m)}$, and $\boldsymbol{W}$, a uniform prior on $\boldsymbol{\Psi}^{(m)}$, and learn MLEs for other parameters. Thus, the goal of our variational algorithm is

to learn optimal variational parameters $\boldsymbol{\lambda}_i = \{\boldsymbol{\nu}_{c_i}, \boldsymbol{\mu}_{b_i}, \boldsymbol{\Sigma}_{b_i}, \{\boldsymbol{\nu}_{z_{im}}, \boldsymbol{\mu}_{f_{im}}, \boldsymbol{\Sigma}_{f_{im}}\}_{m=1}^{M}\}$ for each individual i, as well as a point estimate $\hat{\boldsymbol{\Theta}}$ for model parameters. In practice, we optimize Cholesky decompositions of covariance matrices.

We again use the automatic differentiation package *autograd* in Python to compute exact gradients in order to optimize the lower bound, since the lower bound has an analytic closed-form expression. We again use stochastic variational inference, so that at each iteration of the algorithm, we optimize the local variational parameters for a mini-batch of patients using exact gradients, and then update global parameters with a stochastic gradient (Hoffman et al., 2013). To set the learning rate we use RMSProp, which adaptively allows for a different learning rate for each parameter (Hinton et al., 2012).

2.7 Multivariate Trajectory Model Empirical Study

In this section we describe our experimental setup and results on our dataset.

2.7.1 Dataset

Our dataset contains laboratory values from 44,519 patients with stage 3 CKD or higher extracted from the Duke University Health System EHR. IRB approval (#Pro00066690) was obtained for this work. This data is a superset of the dataset for the joint modeling in Section 2.5.1, the difference being that for this work we only filtered to patients who had at least five recorded values for serum creatinine (rather than ten).

We chpse to model five additional related lab values that have important clinical significance for CKD. The first, serum albumin, is an overall marker of health and nutrition. The second, serum bicarbonate, can indicate acid accumulation from inadequate acid elimination by the kidney, and acidosis is a complication commonly treated. The third, serum calcium, can indicate improperly functioning kidneys if

levels are too high, and is monitored and treated. The fourth, serum phosphorus, can indicate phosphorus accumulation due to inadequate elimination by the kidney, and is associated with cardiovascular death and bone disorders. Finally, urine albumin to creatinine ratio (ACR) is a risk factor and cause of kidney failure (we use a log transform following common practice). While all 44,519 patients have at least five eGFR measurements, there are 884, 242, 78, 16159, and 4321 patients who have no recorded values for the other labs, and the median number of measurements for each lab (among patients with at least one) is 14, 8, 11, 12, 3, and 4, respectively.

As a final preprocessing step, since the recorded eGFR values are extremely noisy and eGFR is only a valid estimate of kidney function at steady state, we take the mean of eGFR readings in monthly time bins for each patient. We also do this to the other lab values, to reduce the overall noise in short time spans. Future work will more explicitly model periods of rapid fluctuation and high variance as they may be related to long term risk. In order to align the patients on a common time axis, for each patient we fix $t = 0$ to be their first recorded eGFR reading below 60 mL/min. The baseline covariates used for $\mathbf{x}_i$ were baseline age, race and gender, and indicator variables for hypertension and diabetes, as well as an overall intercept.

2.7.2 Evaluation

After learning a point estimate for the global model parameters during training, they are held fixed. Then, an approximate posterior over the local latent variables is learned for each patient in the held-out test set. Predictions about future lab values are made by drawing samples from the approximate posterior predictive. We compare our method with the method of Schulam and Saria (2015), trained independently to each of the 6 labs. For each test patient, we learn their parameters three times, using data up until times $t = 1$, $t = 2$, and $t = 4$ (years), each time recording the mean absolute error (MAE) of predictions for each lab in future time windows (e.g.

$t \in (1, 2]$, $t \in (2, 4]$, $t \in (4, 8]$, etc). These values are then averaged over all patients in the test set, and finally averaged over 10 cross validation folds to produce Table 2.1. We use the 10 fold cross validation with one-sided, paired t-tests to test for statistically significant improvements in performance.

2.7.3 Results

Table 2.1: Mean absolute errors across all labs from 10 fold cross validation. Bold indicates p-value from one-sided, paired t-test comparing methods was $< .05$. *,**,*** indicate $p < .01$, $< .001$, $< .0001$, respectively.

Use data up to:		$t = 1$				$t = 2$			$t = 4$	
Lab	Model	$(1, 2]$	$(2, 4]$	$(4, 8]$	$(8, 19]$	$(2, 4]$	$(4, 8]$	$(8, 19]$	$(4, 8]$	$(8, 19]$
eGFR	Schulam	8.84	10.36	12.04	13.78	8.82	11.10	13.17	9.33	12.15
	Proposed	**8.76****	**10.18*****	**11.79*****	13.68	**8.67****	**10.99***	13.13	9.34	12.18
Alb.	Schulam	0.59	0.79	1.06	1.49	0.60	0.87	1.27	0.62	0.96
	Proposed	**0.32*****	**0.37*****	**0.43*****	**0.56*****	**0.33*****	**0.42*****	**0.56*****	**0.38*****	**0.52*****
Bicarb.	Schulam	1.91	2.02	2.13	2.27	1.89	2.04	2.20	1.90	**2.12**
	Proposed	1.85	1.98	2.09	2.29	1.88	2.06	2.29	1.94	2.26
Calc.	Schulam	0.71	1.02	1.56	2.78	0.72	1.21	2.28	0.81	1.55
	Proposed	**0.36*****	**0.41*****	**0.49*****	**0.61*****	**0.38*****	**0.47*****	**0.61*****	**0.42 *****	**0.57*****
Phos.	Schulam	1.02	1.28	1.44	1.40	1.11	1.33	1.29	1.12	1.15
	Proposed	**0.66*****	**0.81****	**1.09**	1.62	**0.72*****	**1.01**	1.46	**0.84***	1.27
ACR	Schulam	1.15	1.30	1.45	1.62	1.14	1.32	1.52	1.14	1.37
	Proposed	**0.84*****	**0.96*****	**1.11****	**1.40**	**0.89*****	**1.09***	1.40	**0.99***	1.31

Table 2.1 displays the results of both methods. We set $G = G_p = 10$, but results were not sensitive to this choice. Our proposed method generally outperforms Schulam and Saria (2015), especially when less data is used to make predictions (e.g. $t = 1$). This makes sense, as our method is able to learn correlations between variables to improve predictions. Further, in cases where the methods had to predict future labs for a patient that did not yet have any observed values for that lab, Schulam and Saria (2015) can only predict using baseline covariates, while our multivariate approach could also leverage information from the other related labs. Current clinical practice for managing care of CKD patients uses clinical judgment alone to predict future disease status. Incorporation of a model such as ours to predict eGFR trajectory and other labs would provide a useful tool for providers to assess the risk of future decline in kidney function.

2.8 Discussion

In this paper, we proposed several classes of Gaussian process-based models for predicting future disease trajectory. We developed a joint model for longitudinal and recurrent event data, and applied it to modeling trajectories of kidney function with prediction of adverse cardiac events in patients with CKD. We also proposed a similar model for multivariate clinical markers, with the end goal of improving predictions about a biomarker of interest by leveraging information from other longitudinal variables. We derived scalable stochastic variational inference algorithms to fit our proposed models, allowing us to apply them to large datasets of longitudinal patient data. We showed that our joint model improves predictive performance compared to competitive baselines for each of the longitudinal and survival submodels. Likewise, we showed that our multivariate trajectory model improves significantly improved performance over univariate modeling by sharing information across related longitudinal variables.

An important assumption made in this work is that observations of the disease trajectories considered are missing at random, implying that we do not need to incorporate information about the sampling model. In future work, we will explore models where the data are missing not at random, where it is necessary to explicate assumptions about the missing data mechanism and incorporate it into the trajectory models.

There are many other directions in which we plan to extend this work as well. Future models will be multivariate in both longitudinal markers and in event processes. Inclusion of additional longitudinal variables such as blood pressure, albuminuria, and hemoglobin A1c will be important, since these are well known to be clinically important for monitoring cardiovascular and kidney health. Incorporation of a larger number of longitudinal variables will require care in order to ensure tractability, es-

pecially if we choose to model frequently recorded vitals. Jointly modeling multiple event processes will allow us to learn correlations between different types of events. More flexible models, particularly for the event processes, should improve model performance, for instance using Gaussian Process modulated Poisson processes or Hawkes processes instead of employing the proportional hazards assumption as we do in this work. Given that much of the data recorded in the EHR is in the form of administrative billing codes, future work should incorporate these into the models as well, perhaps in an unsupervised fashion. We also plan to consider models for other diseases, including diabetes and cardiovascular disease, which are frequently comorbid with CKD. Finally, models incorporating additional outcomes such as medical costs, hospitalizations, and patient quality of life are of significant practical interest.

The screenshot in Figure 2.6 shows the rounding tool developed with Duke Connected Care, where our predictions helped flag high risk patients for chart review during rounding sessions. By further refining and deploying a flexible, scalable model such as ours, ACOs around the country can intervene on high-risk patients and realize the potential benefits of precision medicine.

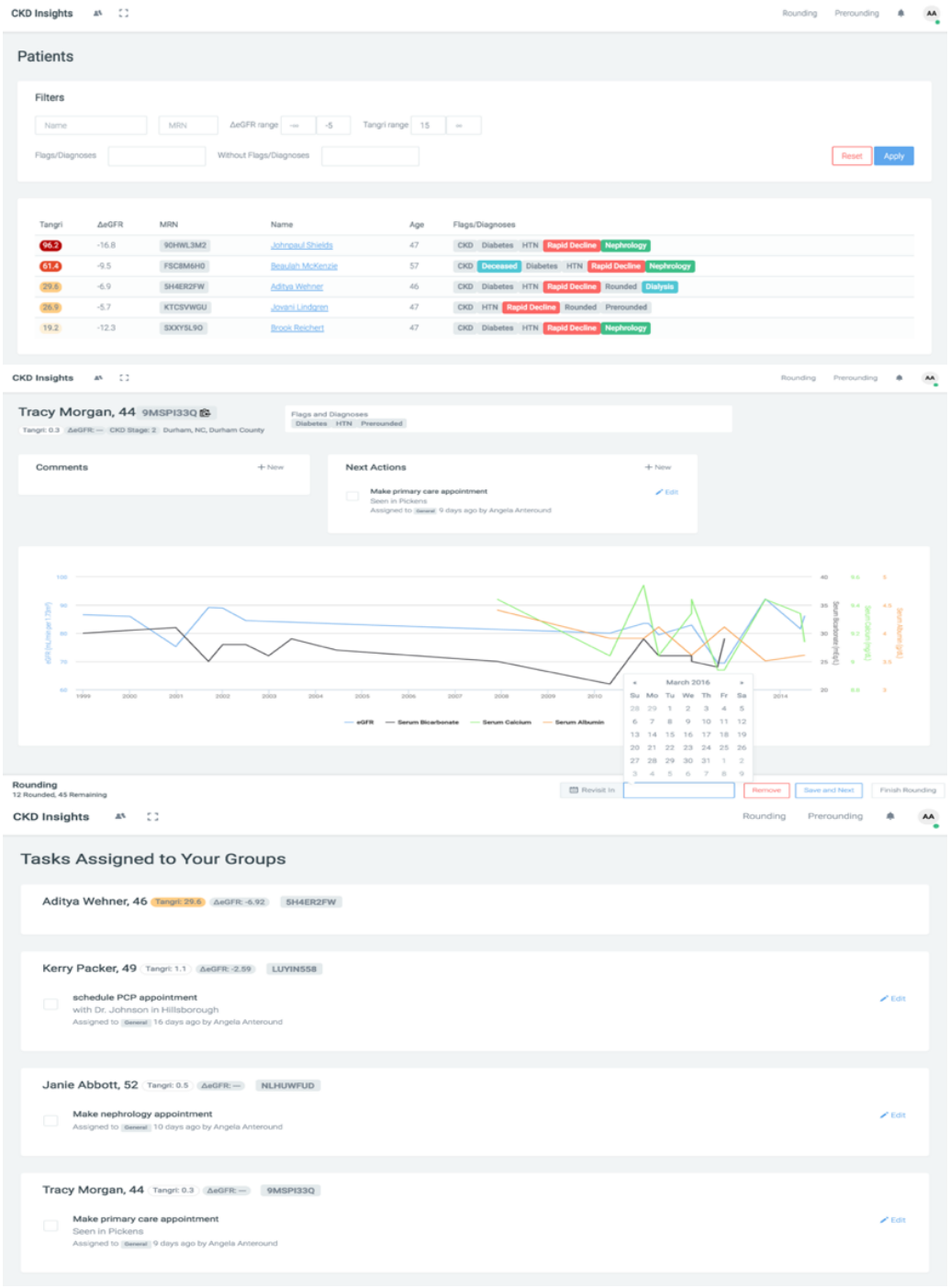

FIGURE 2.6: Snapshots from our CKD rounding application (with synthetic data). The top panel shows a pre-rounding table of patients to be rounded on, along with risk scores and appropriate flags. The middle panel displays patient data and other relevant information so that this patient's care can be efficiently managed and an appropriate intervention made, if applicable. The bottom panel shows a list of tasks assigned to each group present at rounds.

3

Combining Multi-output Gaussian Processes with Deep Learning for Early Diagnosis of Sepsis

3.1 Introduction

Sepsis is a clinical condition involving a destructive host response to the invasion of a microorganism and/or its toxin, and is associated with high morbidity and mortality. Without early intervention, this inflammatory response can progress to septic shock, organ failure and death (Bone et al., 1989). Identifying sepsis early improves patient outcomes, as risk of mortality from septic shock increases by 7.6% for every hour that treatment is delayed after the onset of hypotension (Kumar et al., 2006). Actions such as early fluid resuscitation and administration of antibiotics within hours of sepsis recognition have been shown to improve outcomes (Ferrer et al., 2009). It was also recently shown that timely administration of a 3-hour bundle of care across all patients with sepsis (i.e. blood culture ordered, broad-spectrum antibiotics administered, and lactate measurement taken) was associated with lower in-hospital mortality (Seymour et al., 2017), further emphasizing the need for fast and aggressive treatment. Clearly, early detection of sepsis poses an

important problem in medicine. Unfortunately, early and accurate identification of sepsis remains elusive even for experienced clinicians, as the symptoms associated with sepsis may be caused by many other clinical conditions (Jones et al., 2010).

Despite the difficulties associated with identifying sepsis, data that could be used to inform such a prediction is already being routinely captured in the EHR. To this end, data-driven early warning scores have great potential to identify early clinical deterioration using live data from the EHR. As one example, the Royal College of Physicians developed, validated, and implemented the National Early Warning Score (NEWS) to identify patients who are acutely decompensating (Smith et al., 2013). In particular, NEWS was developed to discriminate patients at high risk of cardiac arrest, unplanned ICU admission, or death. Calculation of these sorts of early warning scores involves comparing a small number of physiological variables (NEWS uses seven) to normal ranges of values to generate a single composite score.

NEWS was previously implemented in our university health system's EHR to improve sepsis detection so that when the score reached a defined trigger, a patient's care nurse was alerted to potential clinical deterioration. However, a major problem with NEWS and other related early warning scores is that they are typically broad in scope and were not developed to target a specific condition such as sepsis, since many unrelated disease states (e.g. trauma, pancreatitis, alcohol withdrawal) can result in high scores. Previous measurements revealed 63.4% of the alerts triggered by the NEWS score at Duke University Hospital were cancelled by the care nurse, suggesting breakdowns in the training and education process, low specificity, and high alarm fatigue. Despite the obvious limitation of using only a small fraction of available information, these scores are also overly simplistic in assigning independent scores to each variable, ignoring both the complex relationships between different physiological variables and their evolution in time, as only the most recent value is used. It should not be surprising that implementation of such scores in clinical

practice results in high alarm fatigue.

3.1.1 Early Warning Scores and Machine Learning for Clinical Deterioration

There is a large body of work on the development and validation of early warning scores to predict clinical deterioration and other related outcomes. In addition to NEWS, the MEWS score (Gardner-Thorpe et al., 2006) and APACHE II score (Knaus et al., 1985) are frequently used to assess overall clinical deterioration. Some specific scores do exist to target sepsis, but in practice these still have high numbers of false alarms. These include the SIRS (systemic inflammatory response syndrome) score, which was part of the original clinical definition of sepsis (Bone et al., 1992), as well as other scores such as SOFA (sepsis-related organ failure assessment) (Vincent et al., 1996) and qSOFA (quickSOFA) (Singer et al., 2016). Despite the initial promise that the newer qSOFA would replace other scores for sepsis detection due to how easy it is to calculate (it only depends on three vital signs), recent work suggests that it severely underperforms existing scores (Churpek et al., 2016; Haydar et al., 2017; Askim et al., 2017).

Motivated in part by the poor performance of these early warning scores, there has been considerable interest in clinical informatics and machine learning on developing more accurate clinical prediction models for predicting sepsis and other types of deterioration. A logistic regression-based approach called the Rothman Index (Rothman et al., 2013) is in widespread use for detecting overall deterioration. Calvert et al. (2016) and Desautels et al. (2016) present related logistic regressions to predict sepsis, using a large number of hand-crafted features. Henry et al. (2015) instead used a Cox regression to predict sepsis from a larger set of clinical time series, although they do not well account for temporal structure, since they simply create feature and event-time pairs from the raw data. Soleimani et al. (2017) improves upon this by using a joint modeling approach similar to the work we presented in

the previous chapter in Section 2.3. Other relevant recent works in machine learning include Yoon et al. (2016) and Hoiles and van der Schaar (2016), as both developed models using clinical time series to predict general deterioration, as observed by admission to the Intensive Care Unit.

3.1.2 Overview of Proposed Modeling Approach

The goal in the work we present in this chapter is to develop a more flexible statistical model that leverages as much available data as possible from patient admissions in order to provide earlier and more accurate detection of sepsis. However, this task is complicated by a number of problems that arise working with real EHR data, some of them particular to sepsis. Unlike other clinical adverse events such as cardiac arrests or transfers to the Intensive Care Unit (ICU) with known event times, sepsis presents a challenge as the exact time at which it starts is generally unknown. Instead, suspicion of sepsis is typically observed indirectly through abnormal labs or vitals, the administration of antibiotics, or the drawing of blood cultures to test for suspected infection. This means the labels in our dataset for when sepsis occurred possess are noisy and not perfectly reliable. Another challenging aspect of our data source is the large degree of heterogeneity present across patient encounters, as we did not exclude certain classes of admissions. More generally, clinical time series data presents its own set of problems, as they are measured at irregularly spaced intervals and there are many missing values, with the missingness frequently informative, as in many cases some labs are taken only if there is suspected problem. Alignment of patient time series also presents an issue, as patients admitted to the hospital may have very different unknown clinical states, with some having sepsis already upon admission. A crucial clinical consideration to be taken into account is the timeliness of alarms raised by the model, as a clinician needs ample time to act on the prediction and quickly intervene on patients flagged as high-risk of being septic. Thus in building a

system to predict sepsis we must consider timeliness of the prediction in addition to other metrics that quantify discrimination and accuracy.

Our proposed methodology for detecting sepsis from multivariate clinical time series overcomes many of these limitations. Our approach hinges on constructing an end-to-end classifier that takes in raw physiology time series data, transforms it through a multi-output Gaussian process (MGP) to a more uniform representation on an evenly spaced grid, and feeds the latent function values through a deep recurrent neural network (RNN) to predict the binary outcome of whether or not the patient will become (or is already) septic. Setting up the problem in this way allows us to leverage the powerful representational abilities of RNNs, which typically requires standardized inputs at uniformly-spaced intervals, for our irregularly spaced multivariate clinical time series. The MGP also serves to de-noise and impute the noisy clinical time series, which have many missing values, while maintaining uncertainty estimates about their true values. As more information is made available during an encounter, the model can dynamically update its prediction about how likely it is that the patient will become septic. When the predicted probability of sepsis exceeds a predefined threshold (chosen to maximize predefined metrics such as sensitivity, positive predictive value, and timeliness), the model can be used to trigger an alarm.

As a motivating example for our work, consider the patient data visualized in Figure 3.1, along with the risk scores generated by our proposed model. This 37 year old female was initially admitted to the hospital for chest pains, and required an invasive cardiac surgery to clear a clot in her lungs. About six days passed between the time when she was admitted and when the surgery was to begin, during which she underwent many preoperative tests but was physiologically stable. However, following surgery she quickly destabilized and was admitted to the Intensive Care Unit (ICU). Shortly after her ICU admission, our model quickly predicted a high

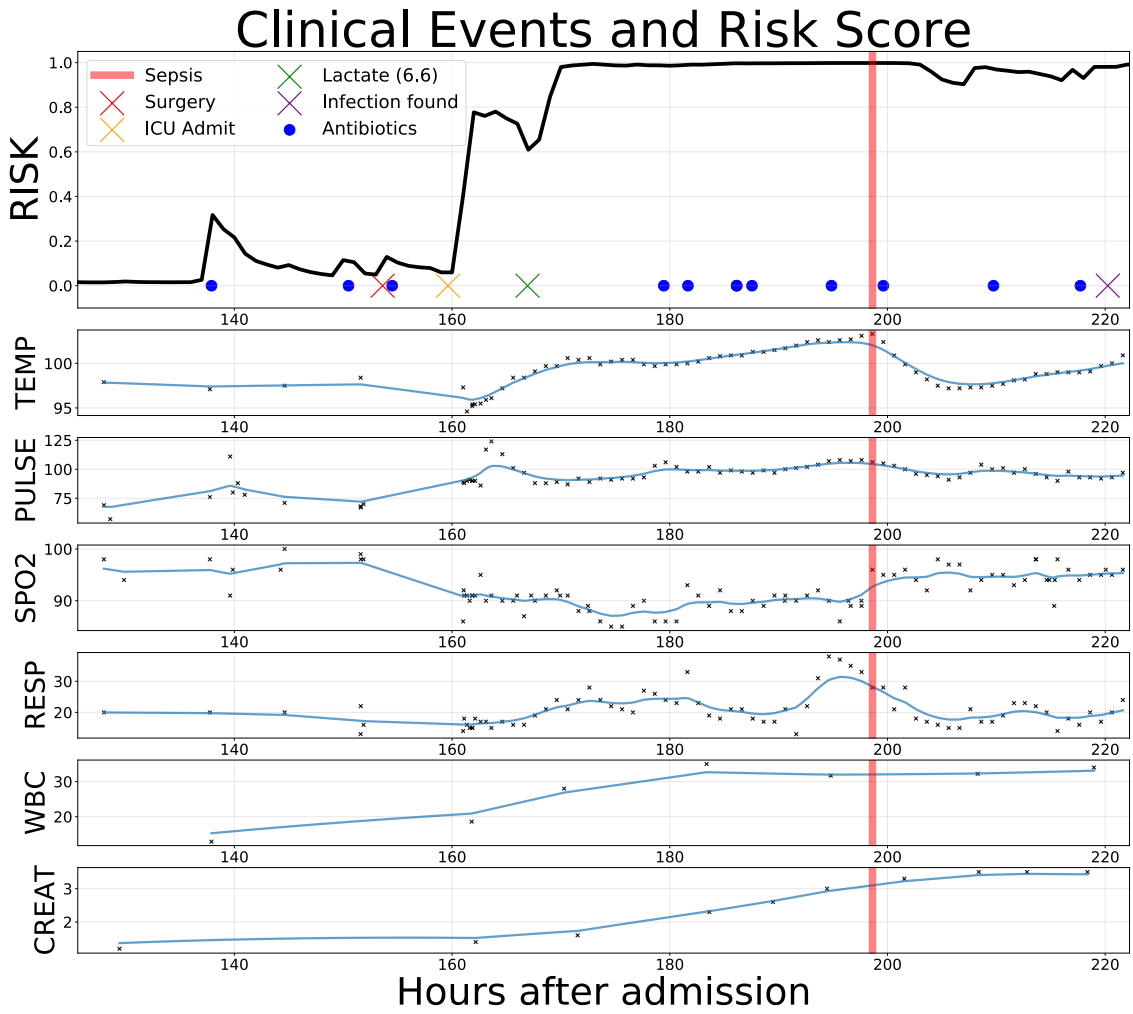

FIGURE 3.1: This patient developed sepsis during a period of rapid deterioration in the ICU following an invasive cardiac surgery. However, our proposed model detected sepsis 17 hours before the first antibiotics were given and 36 hours before a definition for sepsis was met.

risk of sepsis due to her rapid deterioration, and after observing an abnormally high lactate (a common symptom of severe infection), the model became near certain that she was septic. However, it was 17 hours after the model would have detected sepsis that her care team finally started treating her with antibiotics, and another 19 hours until a blood culture was drawn to ascertain the source of the infection. Fortunately, this patient fully recovered and was discharged a week later, although this is not the case for all such missed opportunities. Nonetheless, her care could have been better

managed if her care team was aware of her sepsis earlier and prioritized treating it, which might have led to a faster recovery and shorter hospital stay.

We train our model with real patient data extracted from the Duke University Health System EHR, using a large cohort of heterogeneous inpatient encounters spanning 15 months. We measure predictive performance using several metrics, with an emphasis on obtaining good discrimination while maintaining high precision and low numbers of false alarms. We also validate our model using a "real-time" validation approach that simulates how our model would actually be used in clinical practice. Our overall performance in detecting sepsis is substantially better than the most common early warnings scores from the medical literature, and also offers improvements over competitive baselines. These large gains in performance will translate to better patient outcomes and a lower burden on the overall health system when our model is deployed on the wards in the near future, as our model's predictions will be displayed in a real-time analytics dashboard to be used by a sepsis rapid response team to help detect and improve treatment of sepsis.

3.2 Multi-output Gaussian Processes for Multivariate Clinical Time Series

As we have previously seen, GPs are a common choice for modeling irregularly spaced time series as they are naturally able to handle the variable spacing and differing number of observations per series. Their ability to maintain uncertainty about the variance of the series at each point is important in our setting, since the irregularity and missingness of clinical time series can lead to high uncertainty for variables that are infrequently (or perhaps never) observed, as is often the case. Figure 3.2 provides an example of the type of irregular sampling rates and missingness common in this setting.

In this section we provide a short overview of multi-output Gaussian processes,

where a single Gaussian process serves as a generative model for multivariate time series. As the name suggests, in multi-output Gaussian processes (MGPs), the goal is to learn a function that maps a single input (i.e a time) to multiple outputs (i.e. physiological variables). Note that this approach is fundamentally different than the multivariate disease trajectory model presented earlier in this dissertation in Section 2.6. In that model, a collection of independent univariate GPs were tied together via shared latent variables that characterized their mean functions. This differs from MGPs, where we directly model correlations between the different time series (i.e. between the outputs). See Alvarez et al. (2012) for a detailed overview of multi-output Gaussian processes.

In this initial presentation we focus on multitask Gaussian Processes (MGPs) (Bonilla et al., 2008), which are the simplest possible extension to GPs for handling multiple outputs at each time (we overload abbreviations and also refer to multitask GPs as MGPs; which MGP we refer to should be clear from the context). Let $f_{im}(t)$ be a latent function representing the true values of physiological variable m for patient i at time t. The MGP model places independent GP priors over the latent functions, with a shared correlation function k^t over time. We assume each function has a prior mean of zero, so that the data has been centered. Then, we have:

$$\text{cov}(f_{im}(t), f_{im'}(t')) = \boldsymbol{K}^M_{mm'} k(t, t') \tag{3.1}$$

$$y_{im}(t) \sim \mathcal{N}(f_{im}(t), \sigma^2_m) \tag{3.2}$$

where $y_{im}(t)$ is the actual observed value. Equivalently, the likelihood for $\mathbf{Y}_i \in \mathbb{R}^{T_i \times M}$, a fully observed multivariate time series of M measurements at T_i unique times, is:

$$\text{vec}(\mathbf{Y}_i) \equiv \mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i) \tag{3.3}$$

$$\boldsymbol{\Sigma}_i = \boldsymbol{K}^M \otimes \boldsymbol{K}^{T_i} + \boldsymbol{D} \otimes \boldsymbol{I}, \tag{3.4}$$

where $\mathbf{y}_i$ is a stacked vector of all M longitudinal variables at the T_i observation times, and $\otimes$ denotes the Kronecker product. $\boldsymbol{K}^M$ is a full-rank $M \times M$ covariance ma-

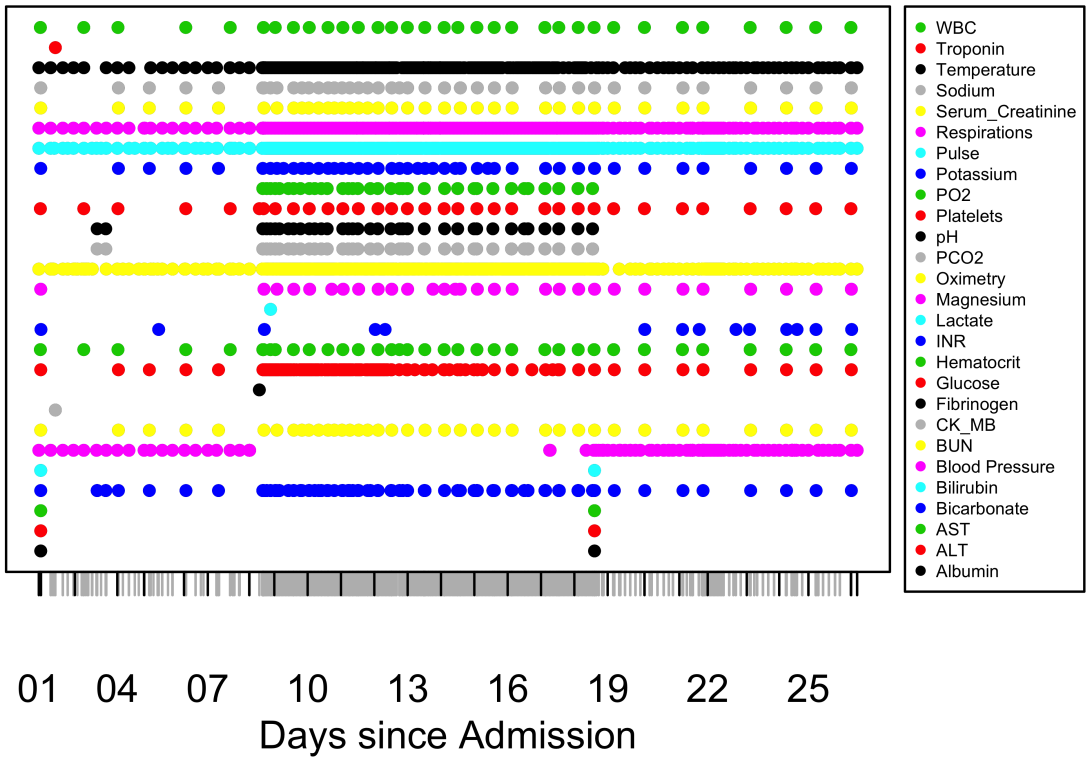

FIGURE 3.2: Data from the same patient encounter highlighted in Figure 3.1, showing when different lab and vital time series variables were measured to highlight their irregular sampling rates. Note the large increase in sampling at about day 8-9; this corresponds to when the patient was transferred to the ICU and was more carefully monitored. Not pictured are the other physiological variables in our dataset that were never measured during this encounter.

trix specifying the relationships among the variables, crucially allowing information from more frequently sampled variables to help improve learning about the variables infrequently (or perhaps never) measured. $\boldsymbol{K}^{T_i}$ is a $T_i \times T_i$ correlation matrix (the variance can be fully explained by $\boldsymbol{K}^M$) for the observation times $\mathbf{t}_i$ as specified by the correlation function k, with parameters η shared across all encounters. In this work we use the Ornstein-Uhlenbeck (OU) kernel function, $k(t, t') = e^{-|t-t'|/l}$, with a single length-scale parameter $\eta = l$. The OU kernel is useful for modeling noisy physiological data, as draws from the corresponding stochastic process are only first-order continuous (Rasmussen and Williams, 2005), and we do not expect the

underlying biological functions to be too smooth. Finally, $\boldsymbol{D}$ is a diagonal matrix of noise variances $\{\sigma_m^2\}_{m=1}^M$. In practice, only a subset of the M series are observed at each time, so the $MT_i \times MT_i$ covariance matrix $\boldsymbol{\Sigma}_i$ only needs to be computed at the observed values. This model is known in geostatistics as the intrinsic correlation model, since the covariance between different variables and between different points in time is explicitly separated, and is a special case of the linear model of coregionalization (Wackernagel, 1998).

The MGP can be used as a mechanism to handle the irregular spacing and missing values in the raw data, and output a uniform representation to feed into a black box classifier. To accomplish this, we define $\mathcal{X}$ to be a set of evenly spaced points in time (e.g. every hour) that will be shared across all encounters. For each encounter, we denote a subset of these points by $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iX_i})$, so that $x_{ij} = x_{i'j}$ if both series are at least x_{ij} hours long. The MGP provides a posterior distribution for the $X_i \times M$ matrix $\mathbf{Z}_i$ of latent time series values at the grid times $\mathbf{x}_i$ within this encounter, while also maintaining uncertainty over the values. If we let $\mathbf{z}_i = \text{vec}(\mathbf{Z}_i)$, this posterior is also normally distributed with mean and covariance given by:

$$\boldsymbol{\mu}_{z_i} = (\boldsymbol{K}^M \otimes \boldsymbol{K}^{X_i T_i}) \boldsymbol{\Sigma}_i^{-1} \mathbf{y}_i \tag{3.5}$$

$$\boldsymbol{\Sigma}_{z_i} = (\boldsymbol{K}^M \otimes \boldsymbol{K}^{X_i}) - (\boldsymbol{K}^M \otimes \boldsymbol{K}^{X_i T_i}) \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{K}^M \otimes \boldsymbol{K}^{T_i X_i}) \tag{3.6}$$

where $\boldsymbol{K}^{X_i T_i}$ and $\boldsymbol{K}^{X_i}$ are correlation matrices between the grid times $\mathbf{x}_i$ and observation times $\mathbf{t}_i$ and between $\mathbf{x}_i$ with itself, as specified by the correlation function k. The set of MGP parameters to be learned are thus $\boldsymbol{\theta} = (K^M, \{\sigma_m^2\}_{m=1}^M, \eta)$, and in this work we assume that they are shared across all encounters. The structured input $\mathbf{Z}_i$ can then serve as a standardized input to a downstream black box classification model, where the raw time series data has been interpolated and missing values imputed.

Many other related works have also utilized multi-output and multitask Gaussian

processes in modeling multivariate physiological time series. For instance, Ghassemi et al. (2015) and Durichen et al. (2015) used a similar multitask GP model to ours, but instead focused more on forecasting of vitals to predict clinical instability, whereas our task is a binary classification to identify sepsis early. Cheng et al. (2017) used a slightly more sophisticated form of multi-output GP, but again the focus was on accurate forecasting of future time series values rather than predicting an event.

3.3 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a class of models from deep learning that have found widespread success in modeling sequential data. These methods first gained popularity in applications such as speech recognition, handwriting recognition, machine translation, and language modeling, due to their state-of-the-art results. In short, an RNN is a form of neural network with recurrent connections that makes it well suited to modeling sequential data, where the sequences can be of variable length. We provide a short introduction to RNNs in this section; see Lipton et al. (2015) for a more thorough overview.

An RNN model typically considers a set of inputs x_t at discrete times t, and attempts to learn a mapping from inputs x_t to outputs y_t. This is accomplished through the use of a neural network with weights that are shared across time steps. In a simple RNN with a single layer, we have:

$$y_t = \sigma_y(W_y h_t + b_y) \tag{3.7}$$

$$h_t = \sigma_h(W_x x_t + W_h h_{t-1} + b_h), \tag{3.8}$$

where h_t are hidden units at time t, and W and b are weight matrices and bias offset terms. The functions σ_y, σ_h are nonlinear activation functions, generally taken to be hyperbolic tangents or sigmoids. This model can be made "deep" by stacking this RNN module into multiple layers so that there are multiple sets of hidden units at

each time. This allows it to better model more complex functions and learn more flexible representations of the input. The hidden units at a given layer serve as inputs to the next layer, until the outputs are reached at the final layer. More explicitly:

$$y_t = \sigma_y(W_y h_t^L + b_y) \tag{3.9}$$

$$h_t^L = \sigma_h(W_x^L h_t^{L-1} + W_h^L h_{t-1}^L + b_h^L) \tag{3.10}$$

$$\dots$$

$$h_t^2 = \sigma_h(W_x^2 h_t^1 + W_h^2 h_{t-1}^2 + b_h^2) \tag{3.11}$$

$$h_t^1 = \sigma_h(W_x^1 x_t + W_h^1 h_{t-1}^1 + b_h^1), \tag{3.12}$$

where again W denotes weight matrices, b denotes bias vectors, h denotes hidden units. The superscript refers to layer index; in this example there are L. A critical shortcoming of simple RNNs such as this one is that they are notoriously hard to train and have serious issues with vanishing and exploding gradients when trained via gradient-based methods. As such, they have severe problems with learning long-range dependencies between variables that are many time steps apart.

Long Short-Term Memory (LSTM) cells are a more complex form of module that can be used instead that were explicitly designed to alleviate these issues (Hochreiter and Schmidhuber, 1997). Due to the severe problems associated with naive RNNs (i.e. of the form described by Equations 3.9–3.12), LSTM RNNs are a standard baseline as they tend to work well in practice and often give competitive performance compared with more sophisticated architectures. Unlike naive RNNs, RNNs built using LSTM cells are able to capture long range dependencies and nonlinear dynamics. The update equations to obtain the next layer of hidden units h_t^l given the previous hidden layer h_t^{l-1} (or the input x_t) and the previous hidden unit h_{t-1}^l

in the same layer are given by:

$$g_t^l = \phi(W_{gx}^l h_t^{l-1} + W_{gh}^l h_{t-1}^l + b_g^l) \tag{3.13}$$

$$i_t^l = \sigma(W_{ix}^l h_t^{l-1} + W_{ih}^l h_{t-1}^l + b_i^l) \tag{3.14}$$

$$f_t^l = \sigma(W_{fx}^l h_t^{l-1} + W_{fh}^l h_{t-1}^l + b_f^l) \tag{3.15}$$

$$o_t^l = \sigma(W_{ox}^l h_t^{l-1} + W_{oh}^l h_{t-1}^l + b_o^l) \tag{3.16}$$

$$s_t^l = g_t^l \odot i_t^l + s_{t-1}^l \odot f_t^l \tag{3.17}$$

$$h_t^l = \phi(s_t^l) \odot o_t^l. \tag{3.18}$$

In these equations σ denotes an element-wise application of the sigmoid function, ϕ is the hyperbolic tangent function and $\odot$ denotes Hadamard (element-wise) products. The intermediary g, i, f, o variables are known as gates; i, f, o are input, forget, and output gates. Informally, the gates limit how much information is passed onward. s is an additional hidden unit that functions as a sort of "memory" to allow the model to better retain information from the past.

Recently, RNNs of many varieties have become popular in modeling clinical time series, as they are able to learn complex nonlinear functions of their input without the need for extensive domain knowledge or feature engineering. This better allows for learning expressive representations and discovering unforeseen structure than methods that rely on hand-crafted features. Lipton et al. (2016a) was one of the first applications of RNNs to medical time series, and they used them to predict diagnosis codes given physiological time series from the ICU. There is a large and quickly growing body of work on developing new RNN architectures and models, and applying them to clinical tasks. Choi et al. (2016b) use Gated Recurrent Unit RNNs to predict onset of heart failure using categorical time series of billing codes, and Zhengping et al. (2016) and Lipton et al. (2016b) investigate patterns of informative missingness in physiological ICU time series with RNNs.

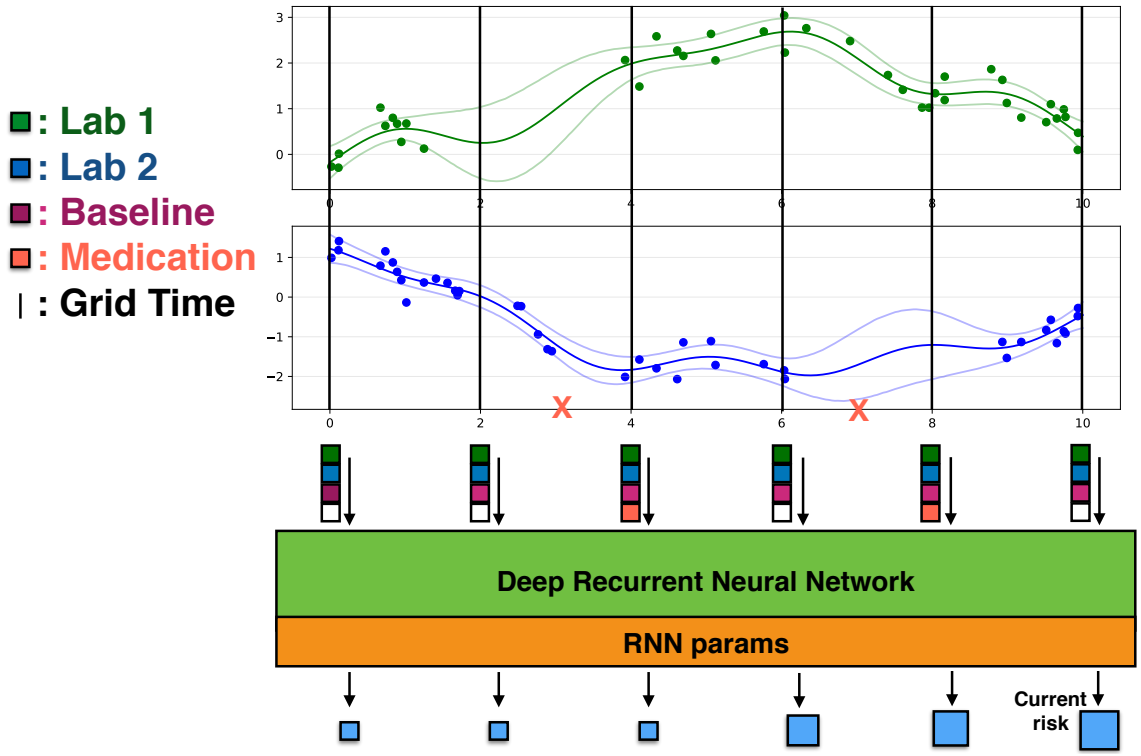

FIGURE 3.3: Schematic for the overall method. At each grid time, latent function values from the MGP model for the clinical time series are fed into the RNN, along with baseline covariates and indicators for medications. The RNN then outputs a probability, which is interpreted as the risk that the patient currently has (or will soon develop) sepsis. During learning we optimize an expected RNN loss rather than the normal RNN loss, since these latent function values are unobserved.

3.4 Multitask Gaussian Process-Recurrent Neural Networks

We frame the problem of early detection of sepsis as a multivariate time series classification problem. Given a new patient encounter, the goal is to continuously update the predicted probability that the encounter will result in sepsis, using all available information up until that time. Figure 3.3 shows an overview of our approach. We first introduce some notation, before presenting the details of the modeling framework, the learning algorithm, and the approximations to speed up learning and inference. Though there is an extensive literature on classification of multivariate time series, these approaches largely rely on clustering using some form of ad-hoc distance metric

between series, and then to make a prediction about a new series it is compared to observed clusters (Xing et al., 2012). Our approach is fundamentally different.

We suppose that our dataset $\mathcal{D}$ consists of N independent patient encounters, $\{\mathcal{D}_i\}_{i=1}^N$. For each patient encounter i, we have a vector of baseline covariates available upon admission to the hospital, denoted $\mathbf{b}_i \in \mathbb{R}^B$, such as gender, age, and documented comorbidities. At times $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{iT_i}]$ during the encounter we obtain information about M different types of vitals and laboratory tests that characterize the patient's physiological state, where $t_{i1} = 0$ is the time of admission. These longitudinal values are denoted $\mathbf{Y}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{iM}] \in \mathbb{R}^{T_i \times M}$, with $\mathbf{y}_{im} \in \mathbb{R}^{T_i}$ the vector of recorded values for variable m at each time. In practice, only a small fraction of this complete matrix is observed, since only a subset of the M variables are recorded at each observation time. We make no assumption about how long each encounter may last, so the length of the time series for each encounter is highly variable ($T_i \neq T_{i'}$) and these times are irregularly spaced, with each encounter having a unique set of observation times. Additionally, during the encounter, medications of P different classes are administered at U_i different times (and it is possible for $U_i = 0$). We denote this information as $\mathcal{P}_i = \{(u_{i1}, \mathbf{p}_{i1}), (u_{i2}, \mathbf{p}_{i2}), \ldots, (u_{iU_i}, \mathbf{p}_{iU_i}\}$, with $\mathbf{p}_{ij} \in \{0, 1\}^P$ a binary vector denoting which of the P medications were administered at time u_{ij}. This information is particularly valuable, because administration of medications provides some insight into a physician's subjective impression of a patient's health state by the type and quantity of medications ordered. Finally, each encounter in the training set is associated with a binary label $o_i \in \{0, 1\}$ denoting whether or not the patient acquired sepsis; we go into detail about how this is defined from the raw data later. Thus, the data for a single encounter can be summarized as $\mathcal{D}_i = \{\mathbf{b}_i, \mathbf{t}_i, \mathbf{Y}_i, \mathcal{P}_i, o_i\}$.

We build off the ideas in Cheng-Xian Li and Marlin (2016) to learn a classifier that directly takes the latent function values $\mathbf{z}_i$ at shared reference time points $\mathbf{x}_i =$

$\{x_{ij}\}_{j=1}^{X_i}$ as inputs. The time series for each encounter i in our data can be represented as an MGP posterior distribution $\mathbf{z}_i \sim N(\mu_{z_i}, \Sigma_{z_i}; \theta)$ at times $\mathbf{x}_i$. This information will be fed into a downstream black box classifier to learn the label of the time series.

Since the lengths of each times series are variable, the classifier used must be able to account for variable length inputs, as the size of $\mathbf{z}_i$ and $\mathbf{x}_i$ will differ across encounters. To this end, we turn to deep recurrent neural networks, a natural choice for learning flexible functions that map variable-length input sequences to a single output. In particular, we used the LSTM architecture (Hochreiter and Schmidhuber, 1997), as these classes of RNNs have been shown to be very flexible and have obtained excellent performance on a wide variety of problems.

At each time x_{ij}, a new set of inputs $\mathbf{d}_{ij} = [\mathbf{z}_{ij}^\top, \mathbf{b}_i^\top, \mathbf{p}_{ij}^\top]^\top$ will be fed into the network, consisting of the M latent function values $\mathbf{z}_{ij}$, the baseline covariates $\mathbf{b}_i$, and $\mathbf{p}_{ij}$, a vector of counts of the S medications administered between x_{ij} and $x_{i,j-1}$. Thus, the RNN is able to learn complicated time-varying interactions among the static admission variables, the physiological labs and vitals, and administration of medications.

If the function values $\mathbf{z}_{ij}$ were actually observed at each time x_{ij}, they could be directly fed into the RNN classifier along with the rest of the observed portion of the vector $\mathbf{d}_{ij}$, and learning would be straightforward. Let $f(\mathbf{D}_i; \boldsymbol{w})$ denote the RNN classifier function, parameterized by $\boldsymbol{w}$, that maps the $(M+B+P) \times X_i$ matrix of inputs $\mathbf{D}_i$ to an output probability. Learning the classifier given $\mathbf{z}_i$ would involve learning the parameters $\boldsymbol{w}$ of the RNN by optimizing a loss function $l(f(\mathbf{D}_i; \boldsymbol{w}), o_i)$ that compares the model's prediction to the true label o_i. However, since $\mathbf{z}_i$ is a random variable, this loss function to be optimized is itself a random variable. Thus, the loss function that we will actually optimize is the expected loss $\mathbb{E}_{z_i \sim N(\mu_{z_i}, \Sigma_{z_i}; \theta)}[l(f(\mathbf{D}_i; \boldsymbol{w}), o_i)]$, with respect to the MGP posterior distribution of $\mathbf{z}_i$. Then the overall learning

problem is to minimize this loss function over the full dataset:

$$\boldsymbol{w}^*, \boldsymbol{\theta}^* = \text{argmin}_{w,\theta} \sum_{i=1}^{N} \mathbb{E}_{z_i \sim N(\mu_{z_i}, \Sigma_{z_i}; \theta)}[l(f(\mathbf{D}_i; \boldsymbol{w}), o_i)]. \tag{3.19}$$

Given fitted model parameters $\boldsymbol{w}^*, \boldsymbol{\theta}^*$, when we are given a new patient encounter $\mathcal{D}_{i'}$ for which we wish to predict whether or not it will become septic, we simply take

$$\mathbb{E}_{z_{i'} \sim N(\mu_{z_{i'}}, \Sigma_{z_{i'}}; \theta^*)}[f(\mathbf{D}_{i'}; \boldsymbol{w}^*)] \tag{3.20}$$

as a risk score that can be updated continuously as more information is available. This approach is "uncertainty-aware", as the uncertainty in the MGP posterior for z_i is propagated all the way through to the loss function. Variations on this setup exist by moving the expectation. For instance, moving the expectation inside the classifier function f swaps the MGP mean vector μ_{z_i} in place of $\mathbf{z}_i$ in the RNN input $\mathbf{D}_i$. This approach will be more computationally efficient but discards the uncertainty information in the time series, which may be undesirable in our setting of noisy clinical time series with high rates of missingness.

3.4.1 End-to-End Learning Framework

The learning problem is to learn optimal parameters that minimize the loss in Equation 3.19. Since the expected loss $\mathbb{E}_{z \sim N(\mu_z, \Sigma_z; \theta)}[l(f(\mathbf{D}; \boldsymbol{w}), o)]$ is intractable for our problem setup, we approximate the loss with Monte Carlo samples:

$$\mathbb{E}_{z \sim N(\mu_z, \Sigma_z; \theta)}[l(f(\mathbf{D}; \boldsymbol{w}), o)] \approx \frac{1}{S} \sum_{s=1}^{S} l(f(\mathbf{D}_s; \boldsymbol{w}), o), \tag{3.21}$$

$$\mathbf{D}_s = [\mathbf{Z}_s^\top, \mathbf{B}^\top, \mathbf{P}^\top]^\top, \;\; \text{vec}(\mathbf{Z}_s) \equiv \mathbf{z}_s \sim N(\mu_z, \Sigma_z; \boldsymbol{\theta}) \tag{3.22}$$

where $\mathbf{B}$ and $\mathbf{P}$ are appropriately sized matrices of the baseline covariates and medication counts over time.

The Reparameterization Trick

We need to compute gradients of this expression with respect to the RNN parameters $\boldsymbol{w}$ and the MGP parameters $\boldsymbol{\theta}$. This can be achieved with the reparameterization trick, using the fact that $\mathbf{z} = \boldsymbol{\mu}_z + \boldsymbol{R}\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim N(\mathbf{0}, \boldsymbol{I})$ and $\boldsymbol{R}$ is a matrix such that $\boldsymbol{\Sigma}_z = \boldsymbol{R}\boldsymbol{R}^\top$ (Kingma and Welling, 2014). This allows us to bring the gradients of Equation 3.21 inside the expectation, where they can be computed efficiently. Rather than choose $\boldsymbol{R}$ to be lower triangular so that it can only be computed in $\mathcal{O}(M^3X^3)$ time with a Cholesky decomposition, we follow Cheng-Xian Li and Marlin (2016) and let $\boldsymbol{R}$ be the symmetric matrix square root, as this leads to a scalable approximation to be discussed shortly. Finally, we train our model discriminatively and end-to-end by jointly learning $\boldsymbol{\theta}$ with $\boldsymbol{w}$, as opposed to a two-stage approach that would first learn and fix $\boldsymbol{\theta}$ before learning $\boldsymbol{w}$.

3.4.2 Scaling Computation with the Lanczos Method

The computation to both learn the model parameters and make predictions for a new patient encounter is dominated primarily by computing the parameters of the MGP posterior in Equations 3.5 and 3.6 and then drawing samples for $\mathbf{z}$ from it, as these are of dimension MX (where X is the number of reference time points). To make this computation more amenable to large-scale datasets such as our large cohort of inpatient admissions, we use the Lanczos method to obtain approximate draws from large multivariate Gaussians.

Recall that to draw from a multivariate Gaussian requires taking the product $\boldsymbol{\Sigma}_z^{1/2}\boldsymbol{\xi}$, where $\boldsymbol{\Sigma}_z^{1/2}$ is the symmetric matrix square root and $\boldsymbol{\xi} \sim N(\mathbf{0}, \boldsymbol{I})$. We can approximate this product using the Lanczos method, a Krylov subspace approximation that bypasses the need to explicitly compute $\boldsymbol{\Sigma}_z^{1/2}$ and only requires matrix-vector products with $\boldsymbol{\Sigma}_z$. The main idea is to find an optimal approximation of $\boldsymbol{\Sigma}_z^{1/2}$ in the Krylov subspace $\mathcal{K}_k(\boldsymbol{\Sigma}_z, \boldsymbol{\xi}) = \text{span}\{\boldsymbol{\xi}, \boldsymbol{\Sigma}_z\boldsymbol{\xi}, \ldots, \boldsymbol{\Sigma}_z^{k-1}\boldsymbol{\xi}\}$; this approximation is simply

Input: covariance matrix $\boldsymbol{\Sigma}$, random vector $\boldsymbol{\xi}$, k
$\beta_1 = 0$ and $\boldsymbol{d}_0 = \mathbf{0}$
$\boldsymbol{d}_1 = \boldsymbol{\xi}/||\boldsymbol{\xi}||$
for $j = 1$ to k **do**
    $\boldsymbol{d} = \boldsymbol{\Sigma}\boldsymbol{d_j} - \beta_j \boldsymbol{d_{j-1}}$
    $\alpha_j = \boldsymbol{d}_j^\top \boldsymbol{d}$
    $\boldsymbol{d} = \boldsymbol{d} - \alpha_j \boldsymbol{d}_j$
    $\beta_{j+1} = ||\boldsymbol{d}||$
    $\boldsymbol{d}_{j+1} = \boldsymbol{d}/\beta_{j+1}$
end for
$\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k]$
$\boldsymbol{H} = \text{tridiagonal}(\boldsymbol{\beta}_{2:k}, \boldsymbol{\alpha}_{1:k}, \boldsymbol{\beta}_{2:k})$
Return: $||\boldsymbol{\xi}||\boldsymbol{D}\boldsymbol{H}^{1/2}\boldsymbol{e}_1$ // $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^\top$

Algorithm 2: Lanczos Method to approximate $\boldsymbol{\Sigma}^{1/2}\boldsymbol{\xi}$

the orthogonal projection of $\boldsymbol{\Sigma}_z\boldsymbol{\xi}$ into the subspace. See Chow and Saad (2014) for more details on the use of Krylov methods for sampling multivariate Gaussians. In practice, k is chosen to be a small constant, $k << MX$, so that the $\mathcal{O}(k^3)$ operation of computing the matrix square root of a $k \times k$ tridiagonal matrix can effectively be treated as $\mathcal{O}(1)$. The most expensive step in the Lanczos method then becomes computation of matrix-vector products $\boldsymbol{\Sigma}_z\boldsymbol{d}$. To compute these we use the conjugate gradient algorithm, another Krylov method, and it usually converges in only a few iterations. We also use conjugate gradient when computing $\boldsymbol{\mu}_z$ in Equation 3.5 to approximate $\boldsymbol{\Sigma}^{-1}\mathbf{y}$. Importantly, every operation in both the Lanczos method, detailed in Algorithm 2, and the conjugate gradient algorithm are differentiable, so that it is possible to backpropagate through the entire procedure during training with automatic differentiation.

3.5 MGP-RNN Empirical Study

3.5.1 Data Description

Our dataset consists of 51,697 inpatient admissions from our university health system spanning 18 months, extracted directly from our EHR. After extensive data cleaning, there were $M = 36$ physiological variables, of which 7 are vitals, and 29 are laboratory

values, and they vary considerably in the number of encounters with at least one recorded measurement. At least one value for each of the vital variables is measured in over 99% of encounters, while some labs (e.g. Ammonia, ESR, D-Dimer) are very rarely taken, being measured in only 2-4% of encounters, with most of the rest falling somewhere in the middle. There were $b = 37$ baseline covariates reliably measured upon admission (e.g. age, race, gender, whether the admission was a transfer or urgent, comorbidities upon admission). Finally, we have information on $P = 11$ medication classes, where these classes were determined from a thorough review of the raw medication names in the EHR. The patient encounters range from very short admissions of only a few hours to extended stays lasting multiple months, with the mean length of stay at 121.7 hours, with a standard deviation of 108.1 hours. As there was no specific inclusion or exclusion criteria in the creation of this patient cohort, the resulting population is very heterogeneous and can vary tremendously in clinical status. This makes the dataset representative of the real clinical setting in which our method will be used, across the entire inpatient wards. Before modeling we log transform all M physiological time series variables to reduce the effect of outliers, and then center and scale all continuous-valued inputs into the model. See Table 3.1 for a full variable list of all inputs to our model.

For encounters that ultimately resulted in sepsis, we used a well-defined clinical definition to assess the first time at which sepsis is suspected to have been present. This criteria consisted of at least two consistently abnormal vitals signs, along with a blood culture drawn for a suspected infection, and at least one abnormal laboratory value indicating early signs of organ failure. This definition was carefully reviewed and found to be sufficient by clinicians. Thus each encounter is associated with a binary label indicating whether or not that patient ever acquired sepsis; the prevalence of sepsis in our full dataset was 21.4%.

For encounters that resulted in sepsis, we used a well-defined clinical definition

Table 3.1: List of all input variables used in our experiments to predict early onset of sepsis. Labs and Continuous Vitals are variables comprising the continuous-valued time series that we model with multi-output GPs.

Variable Type	**Variables**
Labs	Albumin, ALT, Ammonia, AST, Bandemia, Bicarbonate, Bilirubin, BUN, CK-MB, Creatine Kinase, CRP, D-Dimer, ESR, Fibrinogen, Glucose, Hematocrit, INR, Lactate, LDH, Magnesium, PCO2, pH, Platelets, PO2, Potassium, Serum Creatinine, Sodium, Troponin, WBC
Continuous Vitals	Systolic, Diastolic, MAP, Pulse, Pulse Oximetry, Respiratory Rate, Temperature
Categorical Vitals	AVPU_Score (binary: alert / other), Supplemental_Oxygen (binary: yes/no)
Medications	Blood Culture, Antibiotics, Benzodiazepines, Chemotherapy, Heparins, Immunosuppressants, Insulins, IV Fluids, Opioids, Steroids, Vasopressors
Baseline Comorbidities	CHF, Valvular, PHTN, PVD, HTN, Paralysis. NeuroOther, Pulmonary, DM, DMcx, Hypothyroid, Renal, Liver, PUD, HIV, Lymphoma, Mets, Tumor, Rheumatic, Coagulopathy, Obesity, WeightLoss, FluidsLytes, BloodLoss, Anemia, Alcohol, Drugs, Psychoses, Depression
Demographics, Other Info	Age, Gender, Race, Transfer Status, Urgent Admission, EmergencyAdmission, Weight at admission, Prior # Sepsis Encounters

to assess the first time at which sepsis is suspected to have been present. The criteria was: at least two persistently abnormal vitals signs (SIRS score of at least 2/4), a blood culture drawn for suspected infection, and at least one abnormal lab indicating early signs of organ failure. Our criteria most closely matches the "Sepsis-2" definition for severe sepsis (Levy et al., 2003). Although a "Sepsis-3" definition was recently released (Singer et al., 2016), it tends to identify sicker patients with higher mortality, compared with "Sepsis-2", and its adoption is not yet standard. In order to identify more patients potentially at risk of sepsis, we used the older definition. However, our methodology is general and could easily be applied to a similar dataset with a different definition for sepsis. The overall rate of sepsis in the dataset was 21.4%, with each encounter associated with a binary label of whether the patient acquired sepsis, along with a time our sepsis definition was met.

3.5.2 Experimental Setup

We train our method to 80% of the full dataset, setting aside 10% as a validation set to select hyperparameters and a final 10% for testing. For the encounters that result in sepsis, we throw away data from after sepsis was acquired, as our clinical goal is to be able to predict sepsis before it happens for a new patient. For non-septic encounters we train on the full length of the encounter until discharge. We choose the shared reference times $\mathcal{X}$ to be evenly spaced at every hour starting at admission, as clinically the desire is for a risk score that will refresh only once an hour.

We compared our method (denoted "MGP-RNN") against several baselines, including several common clinical scoring systems, as well as more complex methods. In particular, we compared our model with the NEWS score currently in use at our hospital, along with the MEWS score and the SIRS score. Each of these scores are based off of a small subset of the total variables available to our methods. In particular, MEWS uses five, NEWS uses seven, and SIRS uses four. These clinical benchmarks all assign independent scores to each variable under consideration, with higher scores given for more abnormal values, although they each use different thresholds and different values.

As a strong comparison to our end-to-end MGP-RNN classifier, we also trained an LSTM recurrent neural network from the raw data alone (denoted "Raw RNN"), with the same number of layers and hidden units as the network in our classifier (2 layers with 64 hidden units per layer). The mean value for each vital and lab was taken in hourly windows, and windows with missing values carried the most recent value forward. If there was no previously observed variable yet in that encounter, we imputed the mean. In addition, we trained an L_2 penalized logistic regression baseline ("PLR") using this imputation mechanism.

We also compare against a simplified version of the end-to-end MGP-RNN frame-

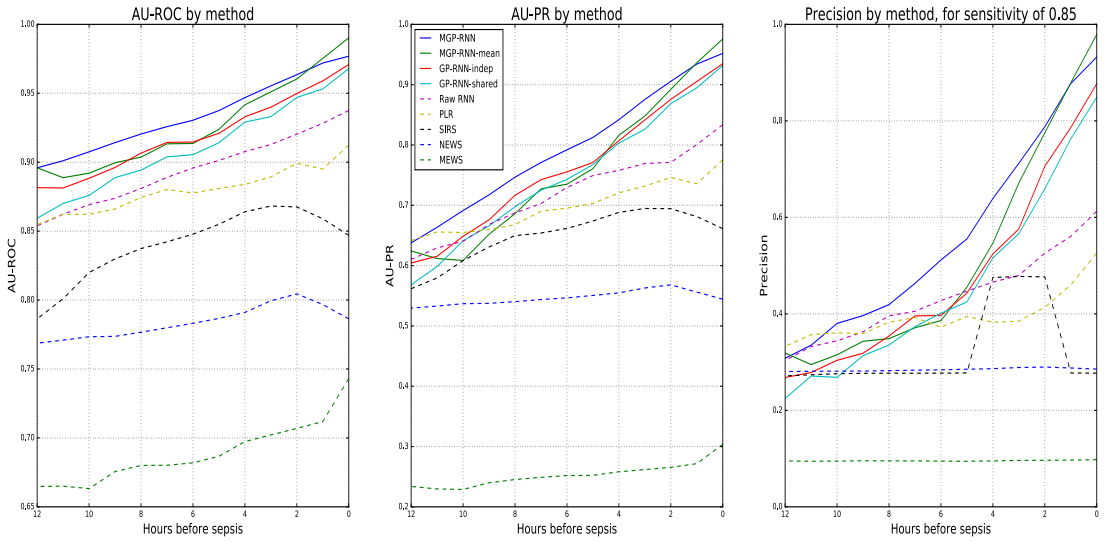

FIGURE 3.4: Left: Area under the Receiver Operating Characteristic curve for each method, as a function of the number of hours in advance of sepsis/discharge the prediction is issued (0-12 hours). Middle: Area under the Precision Recall curve as a function of time. Right: Precision as a function of time, for a fixed sensitivity of 0.85. Methods are all color coded according to middle legend; all GP/MGP-RNN methods are solid lines.

work, (denoted "MGP-RNN-mean") where we replace the latent MGP function values $\mathbf{z}$ with their expectation μ_z during both training and testing, to test the effect of discarding the extra uncertainty information. Finally, to demonstrate the added value of using an MGP instead of independent GPs for each physiological variable, we trained two end-to-end GP-RNN baselines using Equation 3.19, the same loss function as the MGP-RNN. The first, "GP-RNN-shared", is equivalent to an MGP with $K^M = I$, i.e. no covariances across variables, and all variables share the same length-scale in the temporal OU kernel. The second, "GP-RNN-indep", is considerably stronger as it also has no covariances across variables, but allows each GP prior on the latent functions f_m to have its own length scale in its OU kernel, i.e. $l_m \neq l_{m'}$.

To guard against overfitting we apply early stopping on the validation set, and use mild L_2 regularization for all RNN-based methods. We train all models using stochastic gradient descent with the ADAM optimizer Kingma and Ba (2015) using

minibatches of 100 encounters at a time and a learning rate of 0.001. To approximate the expectation in Equation 3.21 we draw ten Monte Carlo samples. We implemented our methods in Tensorflow. On a server with 63GB RAM and 12 Intel Xeon E5-2680 2.50GHz CPUs, the MGP-RNN method takes roughly 10 hours per epoch on the training set, and takes on average 0.3 seconds to evaluate a test case and generate a risk score. All methods converged in a small number of epochs.

3.5.3 Evaluation Metrics

We use several different metrics to evaluate performance. The area under the Receiver Operating Characteristic (ROC) curve (AU-ROC) is an overall measure of discrimination, and can be interpreted as the probability that the classifier correctly ranks a random sepsis encounter as higher risk than a random non-sepsis encounter. We also report the area under the Precision Recall (PR) curve (AU-PR). Importantly, we examine how these metrics vary as we change the window in which we make the prediction to see how far in advance we can reliably predict onset of sepsis.

3.5.4 Results

Our results show that the MGP-RNN classification framework yields clear performance gains when compared to the various baselines considered. It substantially outperforms the overly simplistic clinical scores, and demonstrates modest gains over the RNN trained to raw data, the MGP-RNN-mean method that discards uncertainty information, and the univariate GP baselines.

Figure 3.4 summarizes the results. The four MGP/GP-RNN methods are in solid lines, and the other baselines are dashed. It is clear that these four methods perform considerably better than the other methods, especially in the last four hours prior to sepsis/discharge.

The left and middle panes of Figure 3.4 display the AU-ROC and AU-PR for

each method as a function of the number of hours in advance the prediction is made. Generally the MGP-RNN performs best, followed by the three other MGP/GP-RNN baselines. This is likely because it retains uncertainty information about the noisy time series (unlike MGP-RNN-mean), and can learn correlations among the different physiological variables to improve the quality of the imputation (unlike the GP-RNN methods). As expected, the GP-RNN-indep baseline consistently outperforms the simpler GP-RNN-shared. In addition, the MGP-RNN method appeared to be less prone to overfitting during the training of the RNN, due to the introduced stochasticity in the loss function when compared to the normal RNN loss.

The right pane of Figure Figure 3.4 shows the tradeoff between precision and timeliness for a fixed sensitivity of 0.85 across the methods. It is most useful to evaluate with such a high sensitivity as this is the setting clinicians typically want to use a risk score, in order not to miss many cases. The MGP-RNN performs comparably to the MGP-RNN-mean within a few hours of sepsis, and demonstrates the biggest performance gains from about 3 to 7 hours beforehand. Throughout, it has much higher precision than NEWS, MEWS, and SIRS, especially so in the few hours immediately preceding sepsis. This is a very important clinical point, since clinicians want a method with very high precision and a low false alarm rate to reduce the alarm fatigue experienced with current solutions. Furthermore, being able to detect sepsis even a few hours early might substantially increase treatment effectiveness and improve patient outcomes.

3.6 Extensions to the MGP-RNN

Having introduced the MGP-RNN model and seen its impressive ability to accurately predict sepsis before it occurs, in this section we consider several different mechanisms to improve this base model. We accomplish this in two main ways. The first is by increasing the flexibility of the MGP component, so that it does an even better

job at imputing values for the physiological time series variables. The second is by improving the RNN classifier itself.

3.6.1 Increasing Flexibility of the Multitask Gaussian Process

We increase the flexibility of the original multitask GP in two ways. First, we incorporate medication effects so that the mean of each physiological variable depends on the administration of past drugs. Second, we relax the assumption that the kernel function be separable, and consider a more general covariance function.

Incorporating Medication Effects

We relax the zero mean function assumption for the MGP, and let the mean depend on previous administration of medications. In particular, we let the prior mean function $\mu_m(t)$ for lab/vital m at time t be expressed as:

$$\mu_m(t) = \sum_{p=1}^{P} \sum_{t_p<t} f_{pm}(t - t_p) \tag{3.23}$$

$$f_{pm}(t) = \sum_{l=1}^{L} \alpha_{lpm} e^{-\beta_{lpm} t}, \quad \beta_{lpm} > 0, \tag{3.24}$$

where f_{pm} is a function that specifies the effect medication p has on lab/vital m, and $\{t_p\}$ is the times drug p was given. Our choice of f_{pm} is a flexible family of curves that allows for effects to occur on different length-scales. Each time a new drug is given, the mean function spikes according to f_{pm}. We set $L = 3$, and this seemed to work well in practice.

Sum of Separable Kernel Functions

We next relax the assumption of a separable covariance function (i.e. the covariance function in a multitask GP). Instead, we consider a sum of Q separable covariance functions, each with their own parameters, K_q^M and l_q. The resulting covariance

matrix can be written as

$$\boldsymbol{\Sigma} = \sum_{q=1}^{Q} \boldsymbol{K}_q^M \otimes \boldsymbol{K}_q^T + \boldsymbol{D} \otimes \boldsymbol{I}. \tag{3.25}$$

This is a more flexible family of covariance functions, and no longer forces all output variables to share the same temporal correlation structure (Alvarez et al., 2012; Nguyen and Bonilla, 2014). In this formulation, different types of correlation structure between variables can take place, depending on the time-scale in question. This model is also equivalent to the well-known linear model of coregionalization from geostatistics (Goovaerts, 1997). We found that $Q = 3$ worked well in practice.

3.6.2 Improving the RNN Classifier

We improve the RNN classifier in two ways. First, we use target replication to increase the signal at the end of the series and make learning easier. Second, we use the pattern of missingness in the raw labs/vitals to improve predictions.

Target Replication

Instead of the loss function depending only on the output at the final time step, following Lipton et al. (2016a) we use target replication so that the loss function depends on the outputs of the RNN at multiple time points. This helps to alleviate issues with our imprecise labels for the true time of sepsis, as we can simply label multiple time points near a given time of sepsis. In practice, we use target replication by labelling additional times from 2 hours prior to 6 hours after a sepsis event.

Utilizing Missingness Patterns

We increase the flexibility of our approach by directly modeling the patterns of missing data in the physiological variables, similar to the ideas in Lipton et al. (2016b). To each input vector into the RNN, containing latent physiological function

values from the MGP, baseline covariates, and medications administered, we append a binary vector denoting which labs have been sampled since the last prediction time. This will allow the RNN to model complicated interactions between the missingness patterns in the time series variables, along with the learned values of the variables themselves, and the baseline covariates and meds. This additional information can be very useful, as many labs are only ordered when there is a suspected problem.

3.7 Empirical Study: MGP-RNN Extensions

We use the same dataset as previously introduced in Section 3.5, with the same patient cohort and same set of variables. However, we performed slightly different preprocessing, as we will detail shortly.

We again trained our methods on 80% of the dataset, setting aside 10% for validation to select hyperparameters and the remaining 10% for final evaluation. For all RNNs we used a 2 layer LSTM with 64 hidden units per layer. We used L_2 regularization on the weights and early stopping to guard against overfitting. We train all models using stochastic gradient descent with minibatches of 100 encounters and learning rate of 0.001. Our methods are implemented in Tensorflow.

3.7.1 Case Control Matching

There is an important subtlety to the manner in which we previously trained and validated our methods in Section 3.5. We were comparing predictions about septic encounters shortly before sepsis with predictions about non-septic encounters shortly before discharge. Inclusion of all data for non-septic patients up to discharge is actually not very clinically relevant, as this task would be too easy, since the controls before discharge are likely to be clinically stable.

To make the learning problem more challenging and improve the generalizability of the model, we use a form of case-control matching. The model will instead be

trained to label sepsis encounters around the time of sepsis, and to label control encounters at some time mid-encounter. That is, we are changing the terminal time at which we align patient encounters to be some other time besides discharge for the control encounters. For septic patients we will retain data up until 6 hours after sepsis was acquired, for target replication.

In particular, we first match each sepsis encounter to 4 non-sepsis encounters (this roughly maintains the actual sepsis rate of around 20%) with similar lengths of stay and baseline covariates. Then, we mark a "prediction time" for each control encounter to be at the same fraction of its length of stay as sepsis was during its matched sepsis encounter (e.g. if sepsis occurred at 25% through an encounter, for each matched control we use the time point 25% through the encounter). To train and evaluate our models, we now use data until the time of sepsis plus six additional hours for sepsis cases, and this "prediction time" plus six additional hours for the controls. This is a more realistic problem, since the non-sepsis encounters may not be near discharge now and will be less clinically stable, and the model will better learn what differentiates them from sepsis cases.

3.7.2 Ablation Study and Baseline Methods

We compare a number of variants of our method against several simpler models and clinical baselines. Our base method, denoted "Base MGP-RNN", is our original MGP-RNN with none of the extensions discussed previously. Then, we consider methods where we sequentially add one additional extension at a time. The method denoted "Target Replication" adds target replication to this from Section 3.6.2, using labels from 2 hours prior to sepsis until 6 hours after sepsis. The method "SoS kernel" adds to this by using a sum of $Q = 3$ separable kernels as described in Section 3.6.1. "Medication effect" further adds to this by learning a treatment-response curve for the mean function of the MGP in each dimension, as in Section 3.6.1. Finally,

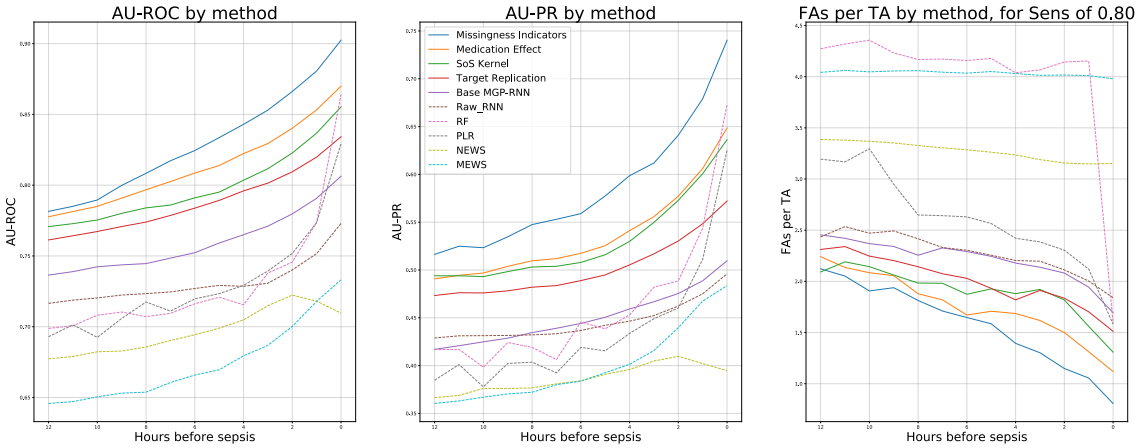

FIGURE 3.5: Results from Matched Lookback validation scheme. Left: AU-ROC as a function of number of hours before sepsis or the matched "prediction time". Middle: AU-PR as a function of hours before sepsis. Right: For a fixed sensitivity of 80%, the number of false alarms per true alarm as a function of hours before sepsis.

"Missingness Indicators" uses all the previous extensions and also feeds indicator vectors for when each physiological variable is measured into the RNN, from Section 3.6.2.

Our strongest baseline method, "Raw RNN", consists of the same network architecture as the MGP-RNN, but instead uses the mean value of each lab/vital in hourly windows, and for periods with missing data the most recent value is carried forward (we use the median for all values of that lab/vital across all encounters if there was no value to carry forward). We also compare with a Lasso logistic regression ("PLR") and random forest ("RF") fit to the same data as the Raw RNN and with the same imputation strategy. Finally, the NEWS and MEWS scores were used as clinical baselines.

3.7.3 Results

We use the area under the ROC curve and the area under the Precision Recall curve as evaluation metrics to compare how each method's performance differs. We also examine the number of false alarms per true alarm for each method, a metric

directly related to precision. We first introduce what we call a "Matched Lookback Validation" scheme and present results from using it.

In this validation strategy, we align matched encounters at either time of sepsis or the "prediction time" for controls, and see how model performance degrades as we make predictions a fixed number of hours in advance of this time. That is, we compare how well the methods discriminate between sepsis and control using all data up through the actual time of sepsis / "prediction time", then up until 1 hour before, and so on, up until 12 hours in advance. This will give a sense for how far in advance we can reliably predict sepsis.

Figure 3.5 shows the results from this validation mechanism. It is clear that the various MGP-RNN methods substantially outperform both the clinical baselines and the other baseline models. The extensions presented to the Base MGP-RNN all improve its performance by a modest margin. The most complete model with all the extensions considered consistently outperformed all other methods for all of the metrics we considered. The number of false alarms per true alarm (right pane of Figure 3.5) is the most clinically useful metric. At 4 hours prior to sepsis, our best model only had about 1.4 false alarms per true alarm, at a very high sensitivity of 80%; compare this to the 2.2 false alarms the base MGP-RNN has, and the 3.2 false alarms that the NEWS score that was previously implemented at Duke Hospital had.

3.8 Realtime Model Validation to Assess Operating Characteristics

A criticism of the previous validation mechanism is that it requires alignment of patients by when their sepsis or "prediction time" is, and this will not actually be known in practice when actually used. In actual clinical practice, a patient encounter begins at admission and the timing of a future outcome is unknown. To alleviate this, we also develop a technique to validate our approach in a more "real-time" setting. For each encounter, we first generate a "real-time" risk score at each hour

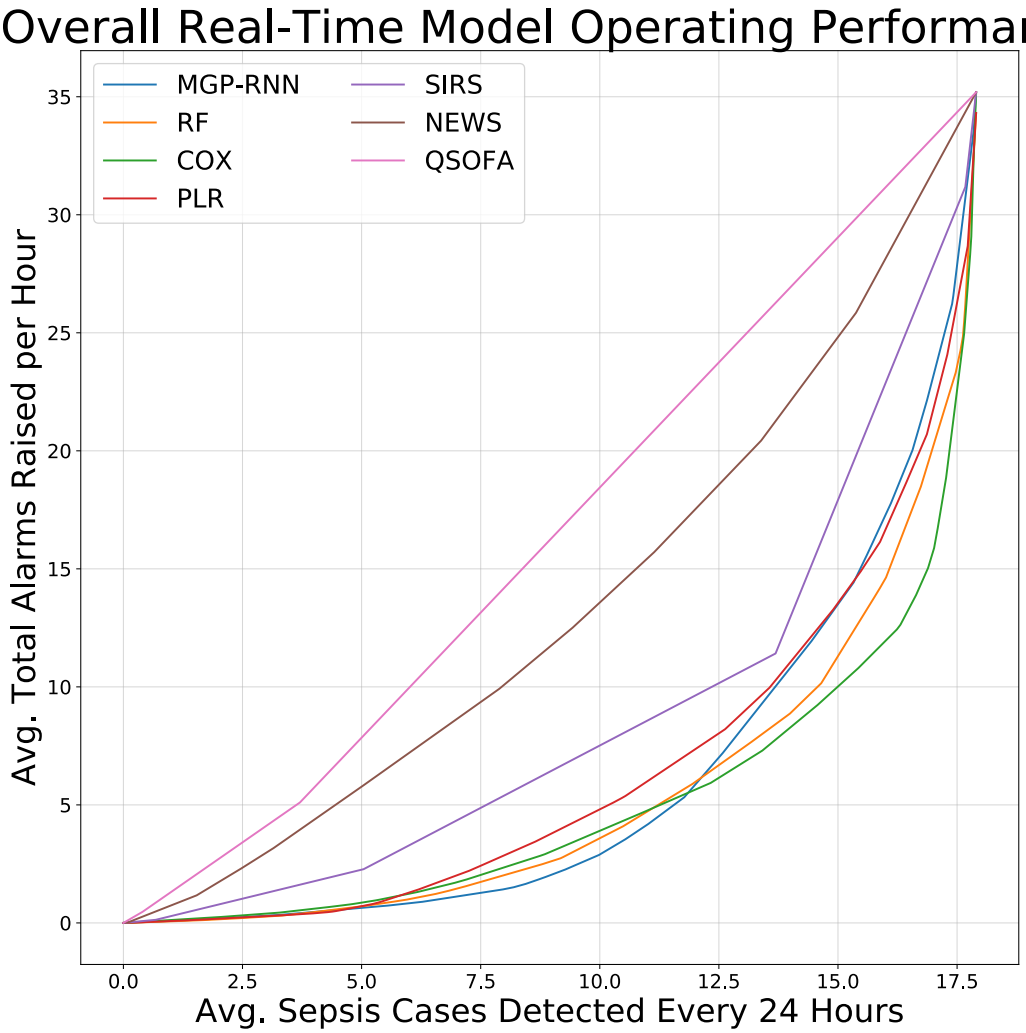

FIGURE 3.6: Expected total number of alarms raised by each method, as a function of expected number of sepsis cases captured every 24 hours. The most clinically useful range is where we restrict the average total alarms per hour to be less than 5.

in time, i.e. using only data up until that point. Then, we bin the encounter into 12 hour long bins, and treat each of these as an independent event. The idea is that in practice a sepsis alert for a patient should not fire more than once every 12 hours. For encounters not resulting in sepsis, every 12 hour window should be a true negative event. For septic encounters, the last 12 hour window preceding sepsis should be a true positive, but all preceding windows should also be true negatives. The rationale is that sounding an alarm prematurely, even if sepsis eventually does

result, is detrimental because in many cases the appropriate clinical response would not be clear. By looking at the full set of risk scores during a full encounter rather than the last few risk scores preceding a septic event, we get a more accurate picture of the operational characteristics of the different methods, to approximate how they might behave if deployed. Treating each 12-hour prediction window as independent events, we can compute performance metrics for each method considered.

Importantly, we explicitly drop the final risk score, so that we are not including any predictions after a clinical definition of sepsis is already met. The reasoning is that in those cases, a prediction model would not provide added value, as the clinical definition would have flagged the patient as septic anyways. In doing so, we have set things up so that each method must make predictions and detect sepsis *before* it actually occurs, and in a clinically meaningful time frame. Thus, sensitivity in this framework refers to the proportion of sepsis cases an alert would correctly identify, in advance of sepsis actually occurring. As might be expected, it will be considerably harder to maintain high precision at high sensitivities now, since there are far more non-septic 12-hour windows than sepsis windows.

We only used real-time validation to compare our best model, with all the MGP-RNN extensions, to a set of baselines. We again use a random forest and penalized logistic regression, and also include a Cox regression. We also compare against the SIRS, NEWS, and qSOFA clinical scores. Figure 3.6 displays the results, showing the average total alarms that would be raised each hour by each method, as a function of the average number of sepsis cases detected every 24 hours (i.e. sensitivity, but rescaled to be more interpretable, as there are on average 17.9 sepsis cases per day in our dataset). Across the board, all 4 machine learning models substantially outperform the 3 clinical scores, which is not surprising. In the low sensitivity range of 0-5 sepsis cases detected per 24 hours (i.e. less than 25%), all the models perform comparably, and would have similar numbers of alerts. For medium sensitivities of

around 10 sepsis cases per 24 hours (i.e. around 55%), the MGP-RNN performs best, and would only raise about 3 total alerts per hour, vs around 4 for RF and Cox and 5 for PLR. Or equivalently, fixing total alerts per hour to 3, MGP-RNN can detect 10 sepsis cases per 24 hours, while RF can detect about 9, Cox about 8.5, and PLR about 7.5. Strangely, at high sensitivities the MGP-RNN no longer performs best. Though alarming, and cause for future investigation, at sensitivities this high it is unlikely for any model to actually achieve a satisfactorily low number of total alerts.

3.9 Conclusions and Clinical Significance

We have presented a novel approach for early detection of sepsis that classifies multivariate clinical time series in a manner that is both flexible and takes into account the uncertainty in the series. On a large dataset of inpatient encounters from our university health system, we find that our proposed method substantially outperforms strong baselines and a number of widespread clinical benchmarks. In particular, our methods tend to have much higher precision and lower rates of false alarm. Importantly, we validated our model and baselines in a real-time manner in order to mimic the way that they would actually be operationalized and used in real clinical practice.

In addition to the initial promise of our approach, there are a number of interesting directions to extend the proposed method. In particular, we could incorporate a clustering component with different sets of MGPs for different latent subpopulations of encounters, to address high heterogeneity across patients. One obvious direction is to improve interpretability of the model so that it is possible to see which inputs at which times contributed the most to the risk score. We are actively investigating this by adding an attention mechanism to the RNN, e.g. along the lines of (Choi et al., 2016a). Exploring other types of flexible black box classifiers such as recurrent variational auto-encoders, e.g. (Chung et al., 2015), may also improve the model's

performance by better accounting for uncertainty in the classifier parameters. Finally, use of additional approximations from the GP literature may further decrease the computational overhead and improve training times.

Due to the importance of this problem in medicine, our work has the potential to have a high impact in improving clinical practice in the identification of sepsis, both at our institution and elsewhere. The underlying biological mechanism is poorly understood, and the problem has historically been very difficult for clinicians. Use of a model such as ours to predict onset of sepsis would significantly reduce the alarm fatigue associated with current clinical scores, and could both significantly improve patient outcomes and reduce burden on the health system. In Figure 3.7 we present a snapshot of an analytics dashboard that is currently being deployed at our hospital system's wards. The tool will be used to display the predictions of our model to predict sepsis to clinicians and nurses on a rapid response team specifically designed to facilitate early detection of sepsis. The application and our model's risk scores will help ensure that early interventions for treatment of sepsis can be started faster for the highest risk patients. Although in this paper our emphasis was on early detection of sepsis, the methods could be used with little modification to detect other clinical adverse events, such as cardiac arrest, cardiogenic shock, or admission to the ICU.

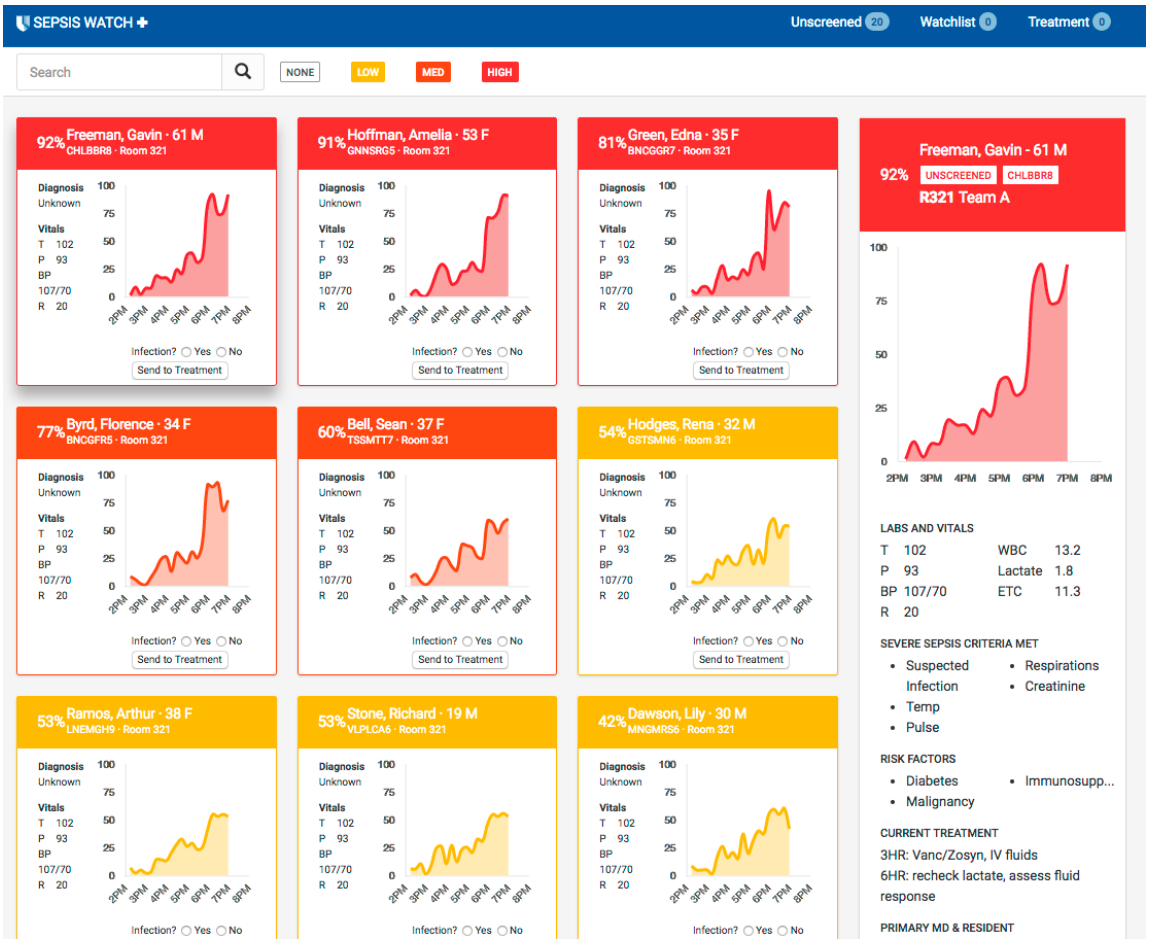

FIGURE 3.7: Screenshot of analytics dashboard (with fake data) that will be used to visualize our model's predictions, to be used by a sepsis rapid response team.

4

Learning Optimal Sepsis Treatments with Multi-output Gaussian Processes and Deep Reinforcement Learning

4.1 Introduction

Sepsis is a poorly understood and life-threatening complication arising from infection, and is both a leading cause in patient mortality (Epstein et al., 2016) and in associated healthcare costs (Torio and Moore, 2016). Early detection is imperative, as earlier treatment is associated with better outcomes (Seymour et al., 2017; Kumar et al., 2006). However, even among patients with recognized sepsis, there is no standard consensus on the best treatment, and exact treatment guidelines do not yet exist. There is a pressing need for personalized treatment strategies tailored to the unique physiology of individual patients.

Before the landmark publication on the use of early goal directed therapy (EGDT) (Rivers et al., 2001), there was no standard management for severe sepsis and septic shock. EGDT consists of early identification of high-risk patients, appropriate cultures, infection source control, antibiotics administration, and hemodynamic opti-

mization. The study compared a 6-hour protocol of EGDT promoting use of central venous catheterization to guide administration of fluids, vasopressors, inotropes, and packed red-blood cell transfusions, and was found to significantly lower mortality. Following the initial trial, EGDT became the cornerstone of the sepsis resuscitation bundle for the Surviving Sepsis Campaign (SCC) and the Centers for Medicare and Medicaid Services (CMS) (Dellinger et al., 2013).

Despite the promising results of EGDT, concerns arose. External validity outside the single center study was unclear, it required significant resources for implementation, and the elements needed to achieve pre-specified hemodynamic targets held potential risks. Between 2014–2017, a trio of trials reported an all-time low sepsis mortality, and questioned the continued need for all elements of EGDT for patients with severe and septic shock (ProCESS Investigators et al., 2014; ARISE Investigators and Anzics Clinical Trials Group, 2014; PRISM Investigators, 2017). The trial authors concluded EGDT did not improve patient survival compared to usual care but was associated with increased ICU admissions (Angus et al., 2015). As a result, they did not recommend it be included in the updated SCC guidelines (Rhodes et al., 2017).

Although the SSC guidelines provide an overarching framework for sepsis treatment, there is renewed interest in targeting treatment and disassembling the bundle (Lewis, 2010b). A recent meta-analysis evaluated 12 randomized trials and 31 observational studies and found that time to first antibiotics explained 96-99% of the survival benefit (Kalil et al., 2017). Likewise, a study of 50,000 patients across the state of New York found mortality benefit for early antibiotic administration, but not intravenous fluids (Seymour et al., 2017). Beyond narrowing the bundle, there is emerging evidence that a patient's baseline risk plays an important role in response to treatment, as survival benefit was significantly reduced for patients with more severe disease (Kalil et al., 2017).

Taken together, the poor performance of EGDT compared to standard-of-care and improved understanding of individual treatment effects calls for re-envisioning sepsis treatment recommendations. Though general consensus in critical care is that the individual elements of the sepsis bundle are typically useful, it is unclear exactly when each element should be administered and in what quantity.

In this chapter, we aim to directly address this problem using deep reinforcement learning. We develop a framework for combining multi-output GPs with deep reinforcement learning, and apply it to clinical data to learn optimal treatments for sepsis. With the widespread adoption of EHRs, hospitals are already automatically collecting the relevant data required to learn such models. However, real-world operational healthcare data present many unique challenges and motivate the need for methodologies designed with their structure in mind. In particular, clinical time series are typically irregularly sampled and exhibit large degrees of missing values that are often informatively missing, necessitating careful modeling. The high degree of heterogeneity presents an additional difficulty, as patients with similar symptoms may respond very differently to treatments due to unmeasured sources of variation. Alignment of patient time series can also be a potential issue, as patients admitted to the hospital may have very different unknown clinical states and can develop sepsis at any time throughout their stay (with many already septic upon admission). Finally, care must taken and assumptions must be made explicit if we hope to learn optimal treatment strategies from observational traces, since we cannot feasibly test the value of our learned strategies.

The main novelty in our approach hinges on the use of a multi-output Gaussian process (MGP) as a preprocessing step that is jointly learned with the reinforcement learning model. Similar to the MGP-RNN model used to predict onset of sepsis in the last chapter, we use an MGP to interpolate and to impute missing physiological time series values used by the downstream reinforcement learning algorithm. The MGP

hyperparameters are learned end-to-end during training of the reinforcement learning model by optimizing an expectation of the standard Q-learning loss. Additionally, the MGP allows for estimation of uncertainty in the learned Q-values. For the model architecture we use a deep recurrent Q-network, in order to account for the potential for non-Markovian dynamics and allow the model to have memory of past states and actions. In our initial experiments utilizing EHR data from septic patients spanning 15 months from the Duke University health system, we found that both the use of the MGP and the deep recurrent Q-network appeared to improve performance over simpler approaches.

4.2 Background on Reinforcement Learning

In this section we provide a brief background on reinforcement learning in the context of healthcare, motivating the new methods we will introduce later in the chapter.

4.2.1 Markov Decision Processes

Reinforcement learning (RL) considers learning policies for agents interacting with unknown environments, and are typically formulated as a Markov decision process (MDP) (Sutton and Barto, 1998). At each time t, an agent observes the state of the environment, $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$, and receives a reward $r_t \in \mathbb{R}$, at which time the environment transitions to a new state s_{t+1}. An MDP is follows the Markov assumption, where we assume the underlying dynamics governing the system are Markov. That is, we assume that there is an unknown transition model of the form $p(s_{t+1}, r_t|s_t, a_t)$ so that the distribution describing the next state s_{t+1} and the current reward r_t are completely determined by the present state s_t and present action a_t. The state space $\mathcal{S}$ and action space $\mathcal{A}$ may be continuous or discrete. In a healthcare setting, the agent might be a decision support tool making recommendations to a provider, the actions are different treatment choices (or no treatment at all), and the

state is some measure of a patient's overall clinical status. The reward is substantially harder to define, as it must numerically quantify precisely what constitutes good vs bad outcomes.

The goal of an RL agent is to select actions in order to maximize its return, or expected discounted future reward, defined as

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}, \tag{4.1}$$

where γ captures tradeoff between immediate and future rewards, with $\gamma = 1$ treating all rewards equally and $\gamma = 0$ being completely myopic. A policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is a map from states to actions, and defines how an agent should act in each possible state. Policies may be deterministic functions, i.e. $\pi(s) = a$, in which case the same action is always taken from the same state. They may also be stochastic, so that each state maps to a probability distribution over possible actions, i.e. $\pi(s) = p(a|s)$. Ultimately, the goal is to learn a good policy that will generally yield high returns, and there are many algorithms to try to accomplish this.

Q-Learning (Watkins and Dayan, 1992) is a model-free off-policy algorithm for estimating the expected return from executing an action in a given state. In model-free RL, we learn a policy directly from observed states, actions, and rewards, without attempting to learn a model for the underlying system dynamics. This has the advantage that it may be computationally cheaper since a model does not have to be learned, and it may be challenging in some settings to learn a good model. However, model-based RL algorithms that learn a model of the environment are more sample-efficient, but may be biased if the learned model is very inaccurate. Q-learning is also an off-policy algorithm. This means that it attempts to learn an optimal policy, even though the data used by the algorithm were generated by another mechanism. In a healthcare setting, we are almost always exclusively limited to off-policy methods,

since the data are generally observational and have already been collected according to some (unknown) policy executed by physicians.

The optimal action value function is the maximum discounted expected reward obtained by executing action a in state s and acting optimally afterwards, defined as

$$Q^*(s,a) = \max_\pi \mathbb{E}[R_t | s_t = s, a_t = a, \pi]. \tag{4.2}$$

Given Q^*, an optimal policy is to act by selecting $\text{argmax}_a Q^*(s,a)$. In Q-learning, the Bellman equation is used to iteratively update the current estimate of the optimal action value function according to

$$Q(s,a) \doteq Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)), \tag{4.3}$$

where we adjust our current estimate of the action value function towards the observed reward plus the maximal Q-value at the next state s'. It has been proven that in finite MDPs with finite $\mathcal{S}$ and $\mathcal{A}$, Q-learning will eventually find an optimal policy, so that the expected return is the maximum achievable. However, in many realistic applications, either the state space, action space, or both are infinite, so that Q-learning cannot be directly applied.

4.2.2 Deep Q-Learning

In Deep Q-learning a deep neural network is used to approximate the Q-values (Mnih et al., 2015), overcoming the issue that there may be infinitely many states if the state space is continuous. Denoting the parameters of the neural network by θ, Q-values $Q(s,a|\theta)$ are now estimated by performing a forward pass through the network. Updates to the parameters can be obtained by minimizing a differentiable loss function,

$$L(s,a|\theta_i) = (r + \gamma \max_{a'} Q(s',a'|\theta_i) - Q(s,a|\theta_i))^2, \tag{4.4}$$

and training is usually accomplished with stochastic gradient descent. However, deep RL models are notoriously difficult to train and often require delicate tuning of hyperparameters to get them to converge.

4.2.3 Partial Observability and Deep Recurrent Q-Networks

A fundamental limiting assumption of Markov decision processes is the Markov property, which is rarely satisfied in real-world problems. In medical applications such as our problem of learning optimal sepsis treatments, it is unlikely that a patient's full clinical state will be measured. A Partially Observable Markov Decision Process (POMDP) better captures the dynamics of these types of real-world environments. An extension of an MDP, a POMDP assumes that an agent does not receive the true state of the system, instead receiving only observations $o \in \Omega$ generated from the underlying system state according to some unknown observation model $o \sim \mathcal{O}(s)$. Deep Q-learning has no reliable way to learn the underlying state of the POMDP, as in general $Q(o, a|\theta) \neq Q(s, a|\theta)$, and will only perform well if the observations well reflect the underlying state. Returning to our medical application, the system state might be the patient's unknown clinical status or disease severity that cannot be directly measured, and our observations in the form of vitals or laboratory measurements offer some insight into the state.

The Deep Recurrent Q-Network (DRQN) (Hausknecht and Stone, 2015) extends vanilla Deep Q-networks (DQN) by using recurrent LSTM layers, which are well known to be able to capture long-term dependencies. LSTM RNN models have frequently been used in past applications to medical time series, as we saw in the last chapter. In our experiments we investigate the effect of replacing fully connected neural network layers with LSTM layers in our Q-network architecture in order to test how realistic the Markov assumption is in our application. It is important to note that using an LSTM or any form of RNN in a DRQN is only an attempt to

circumvent the Markov property, but does not eliminate it. Rather, it allows us to take input states s_t and learn a new representation h_t that depends on the full history of past states and actions, i.e. $h_t = f(s_t, a_{t-1}, s_{t-1}, \ldots, a_1, s_1)$, and then use this representation instead as our notion of state to use in learning an optimal policy.

4.2.4 Related Work on Reinforcement Learning in Healthcare

There has been substantial recent interest in development of machine learning methodologies motivated by healthcare data. However, most prior work in clinical machine learning focuses on supervised tasks; this was the case in the earlier chapters of this dissertation. However, supervised problems rely on known ground truth, and cannot be applied to treatment recommendation unless the assumption is made that past training examples of treatments represent optimal behavior. Especially for our task of learning sepsis treatments, where there is no well established best practices for treatment, this assumption is clearly invalid. Instead, it is preferable to frame the problem using reinforcement learning in order to learn optimal treatment actions from data collected from potentially suboptimal actions.

While deep reinforcement learning has seen huge success over the past few years, only very recently have reinforcement learning methods been designed with healthcare applications in mind. Applying reinforcement learning methods to healthcare data is difficult, as it requires careful consideration to set up the problem, especially in the reward function. Furthermore, it is typically not possible to collect additional data and so evaluating learned policies on retrospective data presents a challenge.

Most related to this paper are Raghu et al. (2016, 2017); Komorowski et al. (2016), who also look at the problem of learning optimal sepsis treatments. We build off of their work by using a more sophisticated network architecture that takes into account both memory through the use of DRQNs and uncertainty in time series imputation and interpolation using MGPs. Other relevant work includes Prasad

et al. (2017), who use a simpler learning algorithms to learn optimal strategies for ventilator weaning, and Nemati et al. (2016), who also use a deep RL approach for modeling ICU heparin dosing as a POMDP with discriminative hidden Markov models and Q-networks. Lastly, Parbhoo et al. (2017) take a different approach and combine model-based RL with kernel methods to learn optimal HIV therapies.

Fundamentally, reinforcement learning in healthcare settings where we are always restricted to off-policy methods is at least as hard as causal inference, and we know that causal inference is extremely challenging. In the simplest setting, causal inference deals with learning the effect a binary treatment at a single point in time. The problem in RL is significantly more complex, as there can be a large number of possible treatments (actions), and there is also a temporal credit assignment problem, where we must determine which actions contributed the most to future rewards or outcomes. There exists a rich set of relevant work from the statistics and causal inference literature on learning so-called dynamic treatment regimes, e.g. Chakraborty and Moodie (2013) and Murphy (2012).

4.3 Multi-Output Gaussian Process Deep Recurrent Q-Networks

We now introduce multi-output Gaussian process deep recurrent Q-networks (MGP-DRQNs), a new reinforcement learning framework for learning optimal treatments from noisy, sparsely sampled, and frequently missing clinical time series data. The fundamental building block is the MGP-RNN model that we introduced in the last chapter.

As we saw in chapter 3, multi-output Gaussian processes (MGPs) are commonly used probabilistic models for irregularly sampled multivariate clinical time series, as they can gracefully handle missing values while maintaining estimates of uncertainty. As in Section 3.6.1, we will use an MGP with the linear model of coregionalization covariance function with an Ornstein-Uhlenbeck base kernels $k^q(t, t') = e^{-|t-t'|/l}$ to

flexibly model temporal correlations in time, with K_q^M specifying the correlation structure across physiological variables associated with kernel k^q. In practice we used $Q = 3$ total kernels, allowing for learning flexible structure on multiple time scales. Given all the MGP kernel hyperparameters $\boldsymbol{\eta}$ shared across all patients, imputation and interpolation at arbitrary times can be computed either using the posterior mean or the full posterior distribution over unknown function values. The MGP can then be combined with an RNN, as we saw in chapter 3. The resulting MGP-RNN model can be learned end-to-end, where the MGP hyperparameters are learned discriminatively, in essence learning an imputation and interpolation mechanism tuned for the supervised task at hand. The stochasticity in this learning procedure introduced from sampling from the MGP additionally acts as a form of regularization, and helps prevent the RNN from overfitting. We can use this same framework, but instead consider a reinforcement learning problem instead of a supervised learning problem.

We will assume a discrete action space, $a \in \mathcal{A} = \{1, \ldots, A\}$. Let $\mathbf{x}$ denote T regularly spaced grid times at which we would like to learn optimal treatment decisions. Given a set of clinical physiological time series $\mathbf{y}$ that we assume to be distributed according to an MGP, we can compute a posterior distribution for $\boldsymbol{z}_t|\mathbf{y}$, the latent unobserved clinical time series values at each grid time.

The loss function we optimize is similar to in normal deep Q-learning, with the addition of the expectation due to the MGP and the fact that we compute the loss over full patient trajectories. In particular, we learn optimal DRQN parameters $\boldsymbol{\theta}^*$ and MGP hyperparameters $\boldsymbol{\eta}^*$ via:

$$\boldsymbol{\theta}^*, \boldsymbol{\eta}^* = \text{argmin}_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathbb{E}\left[\mathbb{E}_{p(\mathbf{z}|\mathbf{y};\eta)}\left\{\frac{1}{T}\sum_{t=1}^{T}(Q_{target}^{(t)} - Q([\boldsymbol{z}_t, \boldsymbol{s}_t]^\top, a; \boldsymbol{\theta}))^2\right\}\right], \tag{4.5}$$

where the t'th target value is

$$Q_{target}^{(t)} = r_t + \gamma \max_{a'} Q([\boldsymbol{z}_{t+1}, \boldsymbol{s}_{t+1}], a'), \tag{4.6}$$

the outer expectation is over training samples, and the inner one is with respect to the MGP posterior for one patient. We concatenate the two separate types of model inputs at time t, with $\boldsymbol{z}_t$ denoting latent variables distributed according to an MGP posterior from other relevant inputs to the model denoted $\boldsymbol{s}_t$, such as static baseline covariates. We go into detail in Section 4.4 on the particular variables included in $\boldsymbol{s}_t$. Figure 4.1 presents a high-level schematic outlining the overall approach of how the MGP and DRQN are tied together.

Architecture

We now discuss in more detail the exact neural network architecture used in our MGP-DRQN model. We base our approach off the Dueling Double-Deep Q-network architecture (Raghu et al., 2017), which combines several modern deep RL architectures.

The Double-Deep Q-network architecture (van Hasselt et al., 2016) helps correct overestimation of Q-values. This can arise in both Q-learning and DQN, since the max operator uses the same set of Q-values to both select and evaluate an action. To do so, we use a second DQN with a different set of parameters $\boldsymbol{\theta}'$ to use to evaluate the target values, and use the original Q-network to select the action. That is, we actually use

$$Q_{target}^{(t)} = r_t + \gamma Q([\boldsymbol{z}_{t+1}, \boldsymbol{s}_{t+1}], \arg\max_{a'} Q([\boldsymbol{z}_{t+1}, \boldsymbol{s}_{t+1}], a'; \boldsymbol{\theta}); \boldsymbol{\theta}'), \tag{4.7}$$

rather than the target given in Equation 4.6.

The Dueling Q-network architecture (Wang et al., 2016) has separate value and advantage streams to separate the effect of a patient being in a good underlying

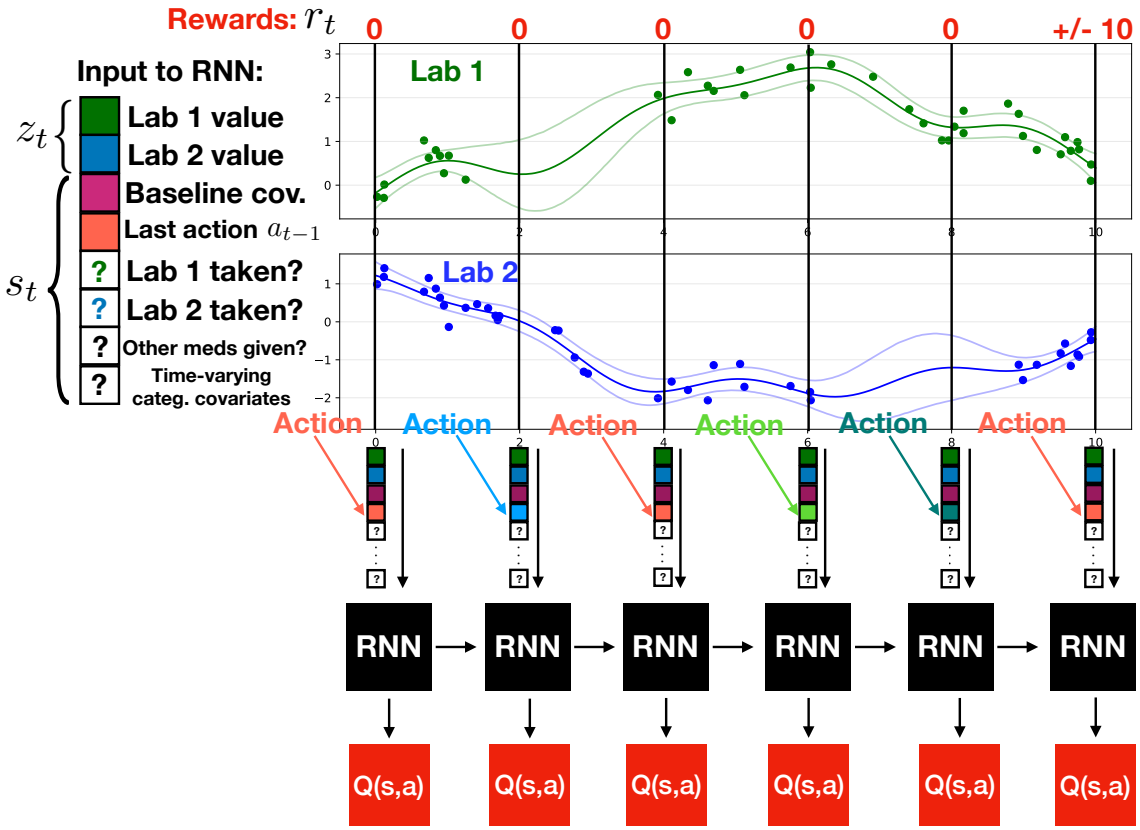

FIGURE 4.1: Schematic outlining the overall approach of the MGP-DRQN. The input variables z_t are latent clinical time series values distributed according to an MGP posterior. The s_t consist of all other inputs to the model. The blue and green dots in the top pane are simulated observations from a two-dimensional time series, and are shown alongside the MGP posterior mean and 95% interval. Areas with fewer observations have higher uncertainty. z_t and s_t together are fed as inputs to the DRQN, which then outputs predictions $Q([z_t, s_t], a)$ for each possible action $a \in \mathcal{A}$.

state from a good action being taken. More explicitly, we let $V(s) = \max_a Q(s, a)$ be the value function associated with state s, and let $A(s, a) = Q(s, a) - V(s)$ be the advantage function, denoting the relative importance of each action. The main idea of the Dueling Q-network is to directly model V and A rather than Q; this helps separate the relative importance of an action from the underlying value of the state. Then, in order to train via deep Q-learning, we can aggregate V and A via

$$Q(s, a; \boldsymbol{\theta}) = V(s; \boldsymbol{\theta}) + A(s, a; \boldsymbol{\theta}). \tag{4.8}$$

Note that in practice V and A will share some parameters in $\boldsymbol{\theta}$, and also have specific parameters of their own.

Finally, we use Prioritized Experience Replay in order to speed learning, so that patient encounters with higher training error will be resampled more frequently (Schaul et al., 2016). That is, when sampling patients to train the RL model with stochastic gradient descent, we do not sample uniformly at random, but instead upweight patients who had high error the last time they were sampled.

For the actual architecture we use 2 LSTM layers with 128 hidden units each that feed to a final fully connected layer with 128 hidden units, before splitting into equally sized value and advantage streams that are finally then projected onto the action space to obtain Q-value estimates. We implemented our methods in Tensorflow using the Adam optimizer with minibatches of 100 encounters sampled at a time, a learning rate of 0.001, L_2 regularization on weights, and leaky ReLU activation functions. We also add an additional regularization penalty to ensure that the learned Q-values do not diverge by penalizing values outside $[-Q_{max}, Q_{max}]$. We use 25 Monte Carlo samples from the MGP for each sampled encounter in order to approximate the expected loss and compute approximate gradients, and these samples and other inputs are fed in a forward pass through the DRQN to get predictions $Q(s, a)$.

4.4 Empirical Study

In this section we first describe the details of our dataset of septic patients before highlighting how the experiments were set up and how the algorithms were evaluated.

4.4.1 Dataset and Preprocessing

Our dataset consists of information collected during 9,255 patient encounters resulting in sepsis from Duke University Hospital, spanning a period of 15 months. We define sepsis to be the first time at which a patient simultaneously had persistently

abnormal vitals (as measured by a 2+ SIRS score, (Bone et al., 1992)), a suspicion of infection (as measured by an order for a blood culture), and an abnormal laboratory value indicative of organ damage. This differs from the new Sepsis-3 definition (Seymour et al., 2016), which has since been largely criticized for its detection of sepsis late in the clinical course (Cortes-Puch and Hartog, 2016). We break the full dataset into 7867 training patient encounters and reserve the remaining 1388 for testing.

We discretize the data to learn actions in 4 hour windows, and limit the data to within 24 hours prior to and 72 hours after the onset of sepsis. We emphasize that the raw data itself is not down-sampled; rather, we use the MGP to learn a posterior for the time series values every 4 hours. Actions for the RL setup consist of 3 treatments commonly given to septic patients: antibiotics, vasopressors, and IV fluids. Antibiotics and vasopressors are broken down into 3 categories, based on whether 0, 1, or 2+ were administered in each 4 hour window. For IV Fluids, we consider 5 discrete categories: either 0, or one of 4 aggregate doses based on empirical quartiles of total fluid volumes. This yields a discrete action space with $3 \times 3 \times 5 = 45$ distinct actions.

Our input data into the RL model is largely the same as in the last chapter for predicting sepsis detection, summarized in Table 3.1. It again consists of 36 longitudinal physiological variables (e.g. blood pressure, pulse, white blood cell count), 2 longitudinal categorical variables, and 38 variables available at baseline (e.g. age, previous medical conditions). 8 medications tangential to sepsis treatment are included as inputs to MGP-DRQN, as well as an indicator for which of the 45 actions was administered at the last time. Additionally, 36 indicator variables for whether or not each lab/vital was recently sampled allows the model to learn from informative sampling due to non-random missingness, following along the lines of the work in the previous chapter to model missing data in Section 3.6.2. In total, there are 165 input observation variables to each of the Q-network models at each time.

Our outcome of interest is mortality within 30 days of onset of sepsis. We use a sparse reward function in this initial work, so that the reward at every non-terminal time point is 0, with a reward of ± 10 at the end of a trajectory based on patient survival/death. Although this presents a challenging credit assignment problem, this allows for data to inform what actions should be taken to reduce chance of death without being overly prescriptive.

4.4.2 Baseline Methods

We use SARSA, an on-policy algorithm, to estimate state-action values for the physician policy. This algorithm is extremely similar to Q-learning, except the goal is to learn the state-action values for the physician policy that actually generated the data, rather than state-action values for an optimal policy. We can also use deep neural networks for function approximation in SARSA, similar to in Q-learning. The Bellman equation for updates in SARSA is almost identical to the Q-learning update in Equation 4.3:

$$Q(s_t, a_t) \doteq Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)), \tag{4.9}$$

the difference being that we use the next state and next action, instead of the next state and optimal next action. After running the SARSA algorithm, we have an estimate of the value of physician policy.

We compare a number of different architectures for learning optimal sepsis treatments. In addition to our proposed MGP-DRQN, we compare against MGP-mean-DRQN, a variant where we move the posterior expectation inside the DRQN loss function, meaning we use the posterior mean of the MGP rather than use Monte Carlo samples from the MGP. We also compare against a DRQN with identical architecture, but replace the MGP with last-one-carried-forward imputation to fill in any missing values, and use the mean if there are multiple measurements. We also

compare against a vanilla DQN, a MGP-DQN, and a MGP-mean-DQN, with an equivalent number of layers and parameters, to test the effect of the recurrence in the DRQN models.

4.4.3 Off-Policy Value Evaluation

Since we only have retrospective data, how can we determine which learned policy is best? In the data, different treatment decisions were made than those that the learned policies would recommend. To estimate the value of each policy in this off-policy setting, we use Doubly Robust Off-policy Value Evaluation (Jiang and Li, 2016) to compute unbiased estimates of each learned optimal policy using our observed off-policy data. In particular, for a given trajectory H, we can compute an unbiased estimate of the value of the learned policy, V_{DR}^{H} using the recursion

$$V_{DR}^{H+1-t} = \hat{V}(s_t) + \rho_t \left(r_t + \gamma V_{DR}^{H-t} - \hat{Q}(s_t, a_t) \right), \tag{4.10}$$

where $\rho_t = \pi_1(a_t|s_t)/\pi_0(a_t|s_t)$, with π_1 the learned policy we wish to evaluate, and π_0 is the physician policy that generated the data. This method combines importance sampling with an approximate MDP model to provide an unbiased and low variance estimate of the quantity V_{DR}^{H}.

For each patient trajectory in the test set we estimate its value using this method, and then average the results. In order to apply this method we also need to estimate the action probabilities of the physician policy, i.e. $\pi_0(a_t|s_t)$, how likely the treating physician was to take each action given a particular state. We can use an MGP-RNN model to estimate these probabilities.

An important caveat and limitation to note is that since our learned policies are deterministic, we often rely only on the estimated reward $\hat{V}(s_t)$ when estimating values. This happens because the importance sampling ratio ρ_t will be 0 when the learned deterministic policy π_1 selects a different action than the one taken by the

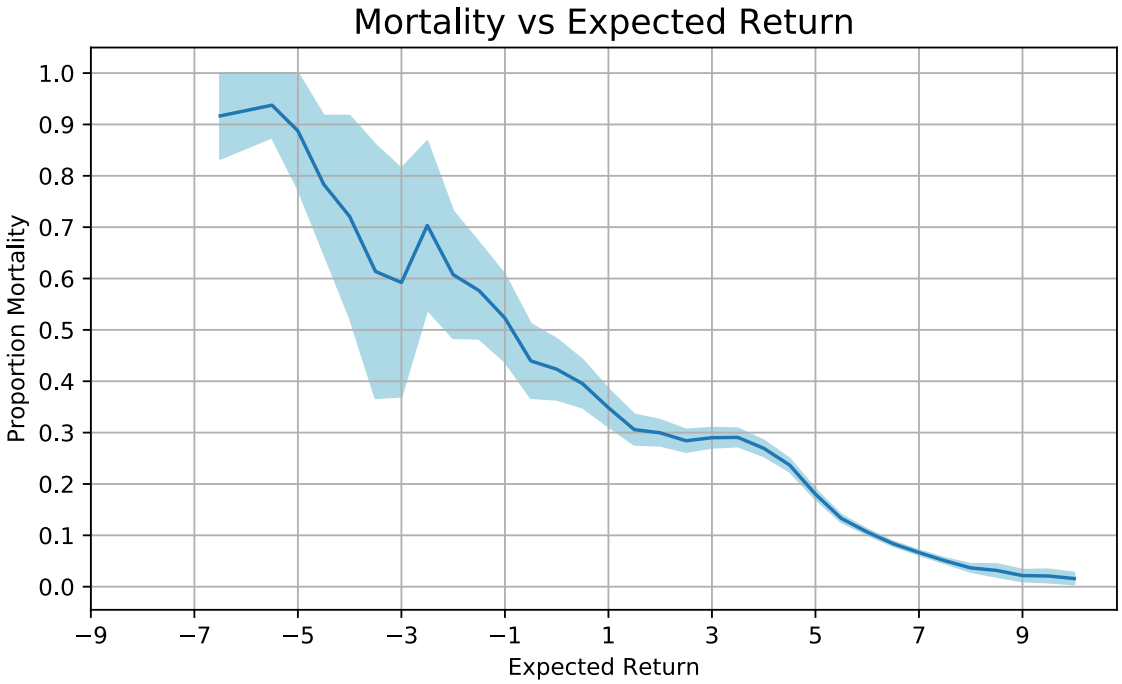

FIGURE 4.2: For the 1388 patients in the test set we show the expected returns as computed by SARSA, against 30-day mortality among patients with similar Q-values. Our model appears to be well calibrated, as higher returns are associated with lower mortality.

clinician. Thus, we are somewhat limited in the accuracy of our value estimates by the accuracy of this estimated reward, and it is hard to quantify our approximation error. For this reason, our quantitative results that we now present are only preliminary, and additional work to improve this evaluation methodology is needed.

4.4.4 Quantitative Results

In Figure 4.2 we show the results of using SARSA to estimate expected returns for the physician policy on the test data. The Q-values appear to be well calibrated with mortality, as patients who were estimated to have higher expected returns tended to have lower mortality. Due to small sample sizes for very low expected returns, the mortality rate does not always monotonically decrease.

We can estimate the potential reduction in mortality a learned policy might have

Table 4.1: Expected returns for the various policies considered. For the 6 reinforcement learning algorithms considered, we estimate their expected returns using an off-policy value evaluation algorithm. Using the results from Figure 4.2, we estimate the potential expected mortality reduction associated with each policy.

Policy	Expected Return	Estimated Mortality
Physician	5.52	13.3 ± 0.7%
MGP-DRQN	**7.51**	**5.1 ± 0.5%**
MGP-mean-DRQN	6.97	6.6 ± 0.4%
DRQN	6.63	8.4 ± 0.4%
MGP-DQN	7.05	6.6 ± 0.4%
MGP-mean-DQN	6.73	7.5 ± 0.4%
DQN	6.09	10.6 ± 0.5%

by computing an unbiased estimate of the policy value, as described in Section 4.3, and then use the results in Figure 4.2. Table 4.1 contains the policy value estimates for each algorithm considered, along with estimated mortality rates. The physician policy has an estimated value of 5.52 and corresponding mortality of 13.3%, matching the observed mortality in the test set of 13.3%. Overall the MGP-DRQN performs best and might reduce mortality by as much as 8%. The DRQN architectures tended to yield higher expected returns, probably because they are able to retain some memory of past clinical states and actions taken. The MGP consistently improved results as well, and the additional uncertainty information contained in the full MGP posterior appeared to do better than the policies that only used the posterior mean.

4.4.5 Qualitative Results

We also qualitatively evaluate the results of the policy from our best performing learning algorithm, the MGP-DRQN, using the test data. In Figure 4.3 we compare the number of times each type of action was actually taken by physicians, and how many times the learned policy selected that action. That is, we break out the actions into the three treatment types: antibiotics, vasopressors, and IV fluids. The MGP-DRQN policy tended to recommend more use of antibiotics and more vasopressors

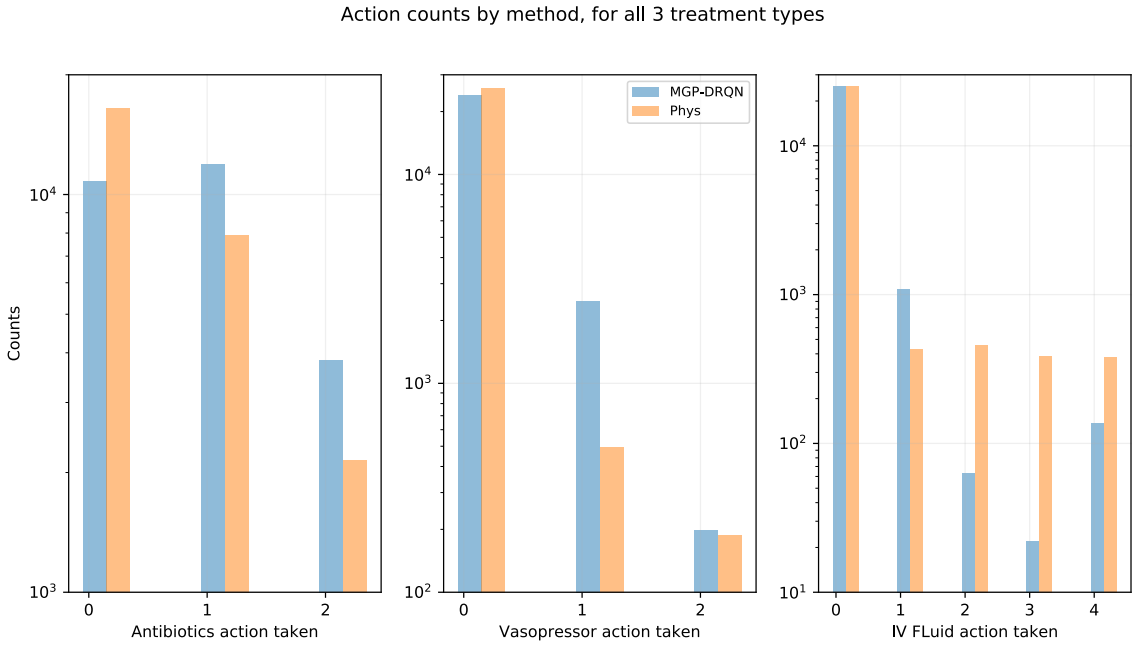

FIGURE 4.3: Comparison of physician actions with the actions that would have been taken by the MGP-DRQN policy, with actions separated according to the 3 types of treatments considered.

than were actually used by physician, while recommending lower use of IV fluids. In Figure 4.4 we again show how often actions are taken, now considering all 45 possible actions.

In Figure 4.5, we show how mortality rates differ on the test set as a function of how different the observed physician action was from what the MGP-DRQN would have recommended. The goal here is to summarize what happens in cases where the MGP-DRQN recommendation and what the physician actually did, vs what happens in scenarios where they differ. For all 3 types of treatments, there appears to be a local minimum at 0 and we observe a V shape, indicating that empirically, mortality tended to be lowest when the clinicians took the same actions that the MGP-DRQN would have. Uncertainty tends to be higher due to smaller sample sizes for situations where there is larger disparity (i.e. there are relatively fewer instances with large disagreement).

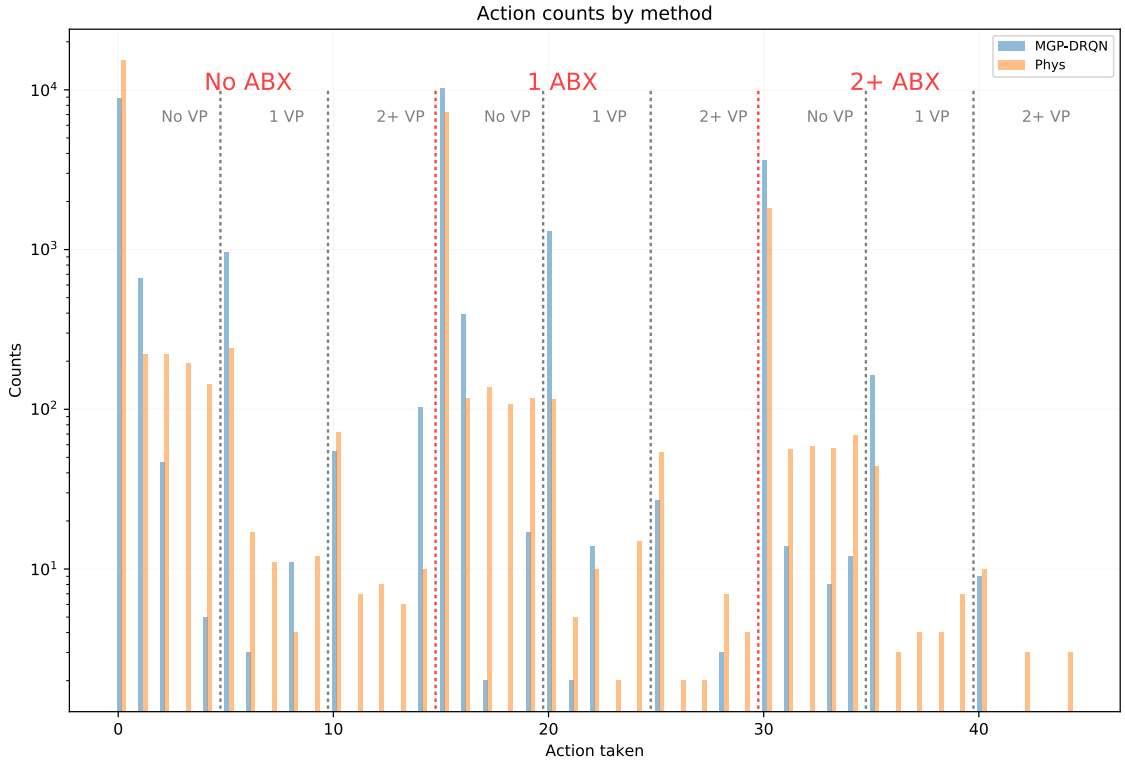

FIGURE 4.4: Comparison of physician actions with the actions that would have been taken by the MGP-DRQN policy. All 45 possible actions are shown.

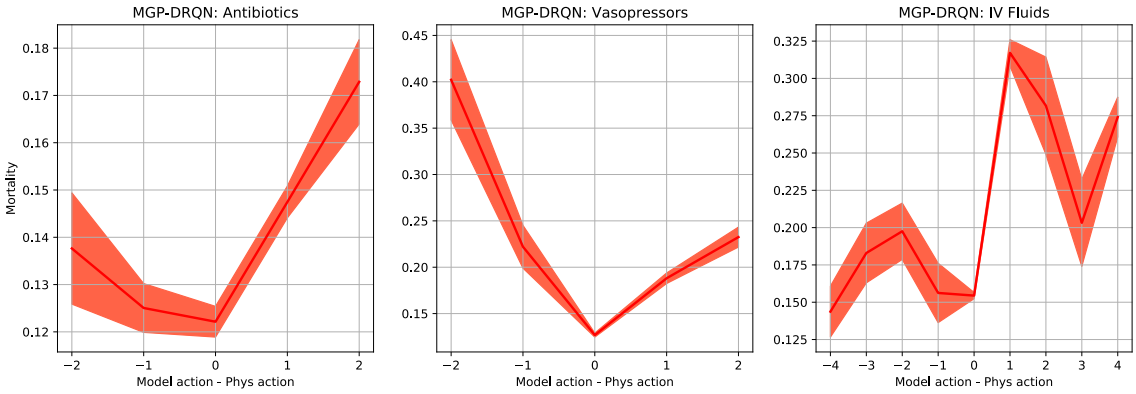

FIGURE 4.5: Empirical mortality rates as a function of how much the MGP-DRQN policy's actions differed from the observed physician actions. Minimal mortality is observed for all 3 treatment types at 0, where the physicians and MGP-DRQN agreed.

Finally, in Figure 4.6 we show clinical data from a sample patient case. In the top pane of the figure we show five representative vital signs and lab measurements to illustrate the patient's clinical status, while the bottom shows both what actions physicians actually took and what actions the model recommended. The patient was admitted to the Emergency Department for altered mental status, and the MGP-DRQN quickly recognizes the need for antibiotics and IV fluids. The patient is admitted to the hospital and around hour 6 the clinical team becomes aware of sepsis. However, antibiotics are not first administered until hour 18, about 16 hours after the model recommended treating with them. After the patient is transferred to the Intensive Care Unit, their white blood cell count continues to rise (a sign of worsening infection) and their blood pressure continues to fall (a sign of worsening shock). By hour 14, the RL model starts and continues to recommend use of vasopressors to attempt to increase blood pressure, but they are not actually administered for about another 16 hours at hour 30. Ultimately, by hour 45 care was withdrawn and the patient passed away at hour 50. Cases such as this one illustrate the potential benefits of using our learned treatment policy in a decision support tool to recommend treatments to providers. If such a tool were used in this situation, it is possible that earlier treatments and more aggressive interventions might have resulted in a different outcome.

4.5 Conclusion

In this paper we presented a new framework combining multi-output Gaussian processes and deep reinforcement learning for clinical problems, and found that our approach performed well in estimating optimal treatment strategies for septic patients. The use of recurrent structure in the Q-network architecture yielded higher expected returns than a standard Q-network, accounting for the non-Markovian nature of real-world medical data. The multi-output Gaussian process also improved

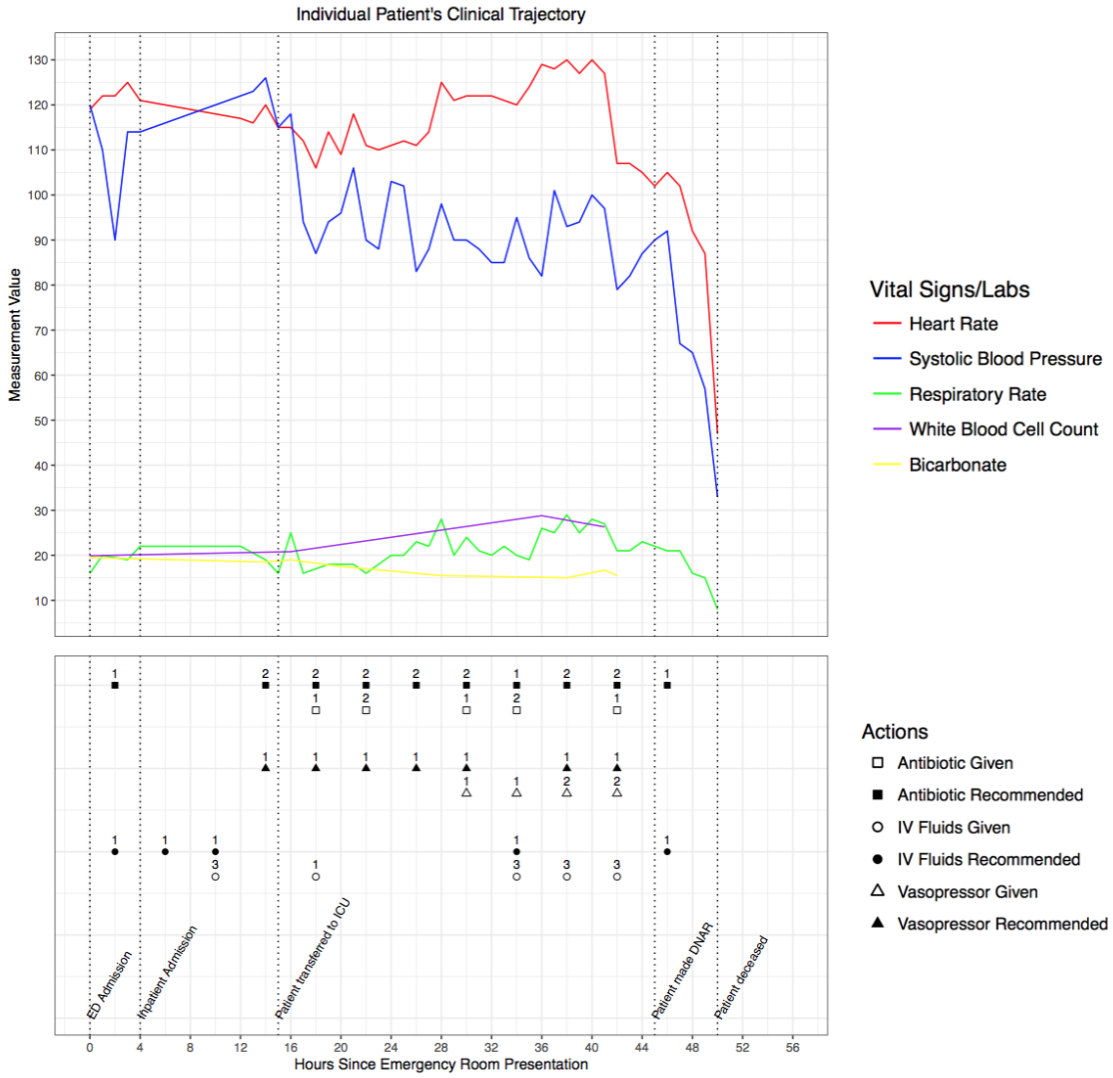

FIGURE 4.6: Top: clinical data from a patient who acquired sepsis, decompensated in the Intensive Care Unit while progressing to septic shock, and ultimately did not survive. Bottom: shaded symbols denote treatments that the learned MGP-DRQN policy would have recommended, while open symbols denote the treatment actions actually taken by physicians caring for this patient.

performance by offering a more principled method for interpolation and imputation, and use of the full MGP posterior improved upon the results from just using the posterior mean.

There are many potential avenues for future work. One promising direction is to investigate the use of more complex reward functions, rather than the sparse rewards used in this work. More sophisticated rewards might take into account clinical targets for maintaining hemodynamic stability, or penalize an overzealous model that recommends too many unnecessary actions. Another important direction is to develop better evaluation methodologies for estimating the value of policies in the off-policy setting, as we discussed in Section 4.4.3. Finally, it would be interesting to extend the MGP preprocessing layer to model-based RL settings. An important advantage of model-based methods, in addition to increased data efficiency, is that we are actually able to simulate counterfactuals since we have learned a model of the underlying dynamics, a feat which is impossible in model-free settings.

In the future, we could include treatment recommendations from our learned policies into our dashboard application we have developed for early detection of sepsis. The treatment recommendations might help providers better care for septic patients after sepsis has been properly identified, and start treatments faster. Our modeling framework is also fairly generalizable, and can easily be applied to other medical applications where there is a need for data-driven decision support tools. In future work we plan to use similar methods to learn optimal treatment strategies for treating patients with cardiogenic shock, and to learn effective insulin dosing regimes for patients on high-dose steroids.

5

Conclusion

In this dissertation, we used Gaussian processes as building blocks in constructing more complex probabilistic models for clinical time series, and applied them to a number of different impactful problems in healthcare. In Chapter 2, we proposed several Gaussian process-based models for predicting future disease trajectory in the context of chronic disease management. We show how a flexible GP model for univariate time series can be extended to jointly model time-to-event data, or to model multivariate time series. In Chapter 3, we presented the MGP-RNN, a method for tying multi-output Gaussian processes for multivariate time series to deep recurrent neural networks. We apply this framework to early detection of sepsis, and show how this approach offers improved predictive performance over a standard deep learning method and other baseline methods. In Chapter 4, we presented preliminary work in progress, and transitioned from predictive to prescriptive modeling. We applied the MGP-RNN to a reinforcement learning problem, where the goal was to learn an optimal treatment regime for sepsis. We showed that the MGP-RNN improves on existing techniques from deep reinforcement learning, and led to higher value policies with potentially lower mortality than the actions taken by physicians.

Despite the large size of modern healthcare datasets in terms of overall sample sizes (i.e. number of patient medical records), there may be considerably less data at an individual level, and so properly accounting for our uncertainty remains important if we intend to use our models to inform decision-making. GPs are especially useful in this regard, as they are probabilistic models that can naturally quantify uncertainty, and we have seen that they perform well in practical settings. In addition, they can act as a form of data augmentation to reduce overfitting in deep learning models, which is especially useful in smaller problems with less data. Future research around ways to build both more flexible and scalable models using GPs remains an active field of research.

5.1 Considerations for Deploying Machine Learning in Healthcare

We conclude this dissertation with a brief discussion around future research directions that will be important in the deployment of machine learning models into real healthcare settings.

5.1.1 Reproducibility, Data Sharing, and Generalizability

Reproducibility in science is a serious problem; in a *Nature* survey of 1,576 researchers across fields, 70% of researchers had tried and failed to reproduce another scientist's experiments, and 52% asserted that there is a "significant crisis" in reproducibility (Baker, 2016). Machine learning is no exception, and many recent papers have addressed some of these issues. Henderson et al. (2017) focus lack of reproducibility in reinforcement learning, while Melis et al. (2017) and Olorisade et al. (2017) address issues in language modeling and text mining, respectively. In a startling number of cases in these papers, simple but well-tuned baseline methods can perform as well or better than more recent complex methods.

Within the machine learning in healthcare community the problem is even more

pronounced due to lack of standardized benchmark datasets. Overwhelmingly, the most commonly used dataset is MIMIC-III, a freely accessible EHR database from ICU patients (Johnson et al., 2016). However, researchers will use different subsets of this database to address distinct questions, and it is not always easy to reproduce their results, even when the source data is publicly available. In Johnson et al. (2017), the authors try to reproduce the results of 28 published studies predicting mortality using MIMIC. Disturbingly, they had significant trouble even building out the same cohort of patients as the published studies, and in half of the experiments the sample size they reproduced was more than 25% off the reported size of the cohort. To alleviate these issues, some work has been done on building benchmark prediction problems from MIMIC to provide a standard comparison (Harutyunyan et al., 2017), but this is not yet widely adopted.

The problem is often even worse in medicine more broadly, as sharing data is not always possible and most healthcare systems are extremely resistant to share or open-source anonymized versions of their data. The one exception to this general rule is in clinical trials, where this is slowly changing due to a federal mandate that data from publicly funded clinical trials be made public. Without the ability to acquire data from other locations, how do we know if a developed model will generalize to a new site? How do we know if a particular method is useful in general prediction tasks, or only in a specific setting? Transfer learning methods may be one promising area of future research, as they offer the ability to learn more generalizable models that can better share relevant information between different sites (Pan and Yang, 2010).

5.1.2 Impactibility: Targeting Highest Impact Patients

A different important problem in deploying machine learning and other forms of clinical prediction models concerns the notion of "impactibility" (Lewis, 2010a). Often

times the information gained from a predictive model is not actionable, especially in settings with limited resources. Given e.g. a patient's risk of acquiring a disease or experiencing an adverse event, what do we do with this knowledge? How do we identify a subset of patients such that collectively we are able to maximize improvement in outcomes across the entire population? In general, our goal should not necessarily be to target the highest-risk patients, but rather the patients who are most likely to benefit from treatment. In some cases, the highest risk patients identified by the model may already be known to providers and receiving adequate treatment that just isn't working, or these patients may be too sick to benefit from treatment. Developing methods to tackle this problem will likely draw from related work in supervised learning, reinforcement learning, causal inference, and counterfactual modeling.

5.1.3 Dataset Drift

An important practical issue not specific to healthcare associated with deploying any predictive model "in the wild" is dataset drift and covariate shift. Although not specific to prediction models in healthcare, healthcare is an application area where these types of problems may have serious consequences. The problem arises when our future test data looks substantially different than the data used to train the model, causing the model to generalize poorly. This can be a huge problem in practice, since the model performance on the training data may be overly optimistic and we may expect it to perform better than it actually does. There are many potential causes, and many variations on the problem. In some cases, the distribution of model inputs x may be different in the test data, while in other cases both inputs x and outputs y may differ in future data. The situation can be even more complex in cases where the model starts to influence human behavior, significantly changing the distribution of new data being generated.

Though some work has been done in this area, e.g. (Quionero-Candela et al.,

2009; Bickel et al., 2009; Lipton et al., 2017), practical questions remain. From an operational perspective, what is the best way to alleviate these potential issues? How frequently should we be retraining our production models? Should more recent data be weighted more heavily? These are all largely open problems. Substantial engineering and automated preprocessing tools may prove necessary to help detect and avert these types of issues in practice.

5.1.4 Interpretability

A final issue we highlight relates to model interpretability, and related notions of explainability and transparency. Problems in these area have recently received a lot of attention, in large part because of the opaqueness and black-box nature of modern deep learning methods. This has motivated many new methods that attempt to explain the predictions of a black-box model. One popular approach, LIME (Local Interpretable Model-Agnostic Explanations) uses a local linear model to craft explanations (Ribeiro et al., 2016). Other methods for post-hoc model interpretation use influence functions (Koh and Liang, 2017) and input gradients (Sundararajan et al., 2017). In other cases, the model itself is constructed in such a way that its predictions are more naturally explainable or interpretable in some sense, e.g. (Lei et al., 2016; Ross et al., 2017; Al-Shedivat et al., 2017; Li et al., 2017). A key problem in general in research in interpretability is the lack of formal definitions for what it even means for a model to be interpretable. To this end, Lipton (2016) and Doshi-Velez and Kim (2017) argue for the need for more rigor in the definition of interpretability.

On a more practical level, interpretability should ultimately be guided by how it relates to overall utility and usability of a model. There may be some situations where black-box predictions without explanation are acceptable; in other cases, explanations about how the model generated its prediction may help increase trust in the system and improve adoption. This also ties in to potential problems in user

interfaces and human computer interaction, around defining the most useful ways to visualize and consume the information that a model produces. There is still substantial work to be done in this overall area.

5.2 Final Thoughts

Machine learning applied to healthcare is a rapidly growing area of research. Many papers are being published very quickly, but often without clinical guidance or input from actual physicians. As a field, it is critical to ensure that clinicians are in the loop during the modeling process, to make sure that we are even solving the right problems. Without understanding the problem domain and the problems clinicians actually face, we cannot begin to solve them.

Clinical decision support systems have been around for some time, and can improve clinical practice in many cases, such as improving prescribing practices and reducing serious medication errors. However, there are few clinical decision support tools in use that directly integrate predictive modeling to improve decision making. It takes careful planning to integrate models into such software tools in a way that is useful to the end users. We need comparative user studies to demonstrate what clinical workflows are most effective to engage end users of a prediction model, rather than overwhelm them. It is our responsibility as those developing new methods to educate the end users, and be explicit about critical modeling assumptions. If we oversell machine learning and do not provide sobering reminders about what our models can and cannot do, we wind up with overhyped systems that suffer from unintended consequences (Cabitza et al., 2017; Chen and Asch, 2017).

Ultimately, the only way to achieve meaningful progress in this space and actually use machine learning to improve healthcare and medicine is through close partnerships between quantitative researchers and practicing physicians. We conclude with thoughts from a recent New England Journal of Medicine perspectives

piece (Obermeyer and Lee, 2017):

> "Algorithms that learn from human decisions will also learn from human mistakes, such as overtesting and overdiagnosis, failing to notice people who lack access to care, undertesting those who cannot pay, and mirroring race or gender biases. Ignoring these facts will result in automating and even magnifying existing problems in our current health system. Noticing and undoing these problems requires a deep familiarity with clinical decisions and the data they produce – a reality that highlights the importance of viewing algorithms as thinking partners, rather than replacements, for doctors. Ultimately, machine learning in medicine will be a team sport, like medicine itself."

Appendix A

Software

A.1 CKD-JM

Code for the disease trajectory models described in Chapter 2 is publicly available on Github at: `https://github.com/jfutoma/CKD-GP`.

A.2 MGP-RNN

Code for the MGP-RNN models described in Chapters 3 and 4 is publicly available on Github at: `https://github.com/jfutoma/MGP-RNN`.

Bibliography

Al-Shedivat, M., Dubey, A., and Xing, E. P. (2017), "Contextual explanation networks," *arXiv preprint arXiv:1705.10301*.

Allen, A. S., Forman, J. P., Orav, E. J., Bates, D. W., Denker, B. M., and Sequist, T. D. (2011), "Primary care management of chronic kidney disease," *J. Gen. Intern. Med.*, 26, 386–92.

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012), "Kernels for vector-valued functions: a review," *Foundations and Trends in Machine Learning*, 4, 195–266.

Amarasingham, R., Patzer, R. E., Huesch, M., Nguyen, N. Q., and Xie, B. (2014), "Implementing electronic health care predictive analytics: considerations and challenges," *Health Aff.*, 33, 1148–54.

Angus, D. C., Barnato, A. E., Bell, D., Bellomo, R., Chong, C. R., Coats, T. J., Davies, A., Delaney, A., Harrison, D. A., Holdgate, A., Howe, B., Huang, D. T., Iwashyna, T., Kellum, J. A., Peake, S. L., Pike, F., Reade, M. C., Rowan, K. M., Singer, M., Webb, S. A., Weissfeld, L. A., Yealy, D. M., and Young, J. D. (2015), "A systematic review and meta-analysis of early goal-directed therapy for septic shock: the ARISE, ProCESS and ProMISe investigators," *Intensive Care Med.*, 41, 1549–1560.

ARISE Investigators and Anzics Clinical Trials Group (2014), "Goal-directed resuscitation for patients with early septic shock," *N. Engl. J. Med*, 371, 1496–1506.

Askim, A., Moser, F., Gustad, L. T., Stene, H., Gundersen, M., Åsvold, B. O., Dale, J., Bjørnsen, L. P., Damås, J. K., and Solligård, E. (2017), "Poor performance of quick-SOFA (qSOFA) score in predicting severe sepsis and mortality: a prospective study of patients admitted with infection to the emergency department," *Scand. J. Trauma Resusc. Emerg. Med.*, 25, 56.

Baker, M. (2016), "1,500 scientists lift the lid on reproducibility," *Nature News*, 533, 452.

Barnes, S., Hamrock, E., Toerper, M., Siddiqui, S., and Levin, S. (2015), "Real-time prediction of inpatient length of stay for discharge prioritization," *J. Am. Med. Inform. Assoc.*, 23, e2–e10.

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., and Escobar, G. (2014), "Big Data In Health Care: Using Analytics To Identify And Manage High-Risk And High-Cost Patients," *Health Aff.*, 33, 1123–1131.

Beam, A. L. and Kohane, I. S. (2016), "Translating artificial intelligence into clinical care," *JAMA*, 316, 2368–9.

Bejnordi, B. E., Veta, M., van Diest, P. J., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A. W. M., and CAMELYON16 Consortium (2017), "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *JAMA*, 318, 2199–2210.

Berger, J. O. (2013), *Statistical decision theory and Bayesian analysis*, Springer Science & Business Media.

Bickel, S., Brückner, M., and Scheffer, T. (2009), "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, 10, 2137–2155.

Bishop, C. (2006), *Pattern recognition and machine learning*, Springer, New York.

Bone, R. C., Fisher, C. J., Clemmer, T. P., Slotman, G. J., Metz, C. A., and Balk, R. A. (1989), "Sepsis syndrome: a valid clinical entity," *Crit. Care Med.*, 17, 389–93.

Bone, R. C., Balk, R. A., Cerra, F. B., Dellinger, R. P., Fein, A. M., Knaus, W. A., Schein, R. M., and Sibbald, W. J. (1992), "Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis," *Chest*, 101, 1644–55.

Bonilla, E. V., Chai, K. M. A., and Williams, C. K. I. (2008), "Multi-task Gaussian Process Prediction," in *Advances in Neural Information Processing Systems*, p. 153160.

Cabitza, F., Rasoini, R., and Gensini, G. F. (2017), "Unintended consequences of machine learning in medicine," *JAMA*, 318, 517–518.

Calvert, J. S., Price, D. A., Chettipally, U. K., Barton, C. W., Feldman, M. D., Hoffman, J. L., Jay, M., and Das, R. (2016), "A computational approach to early sepsis detection," *Comput. Biol. Med.*, 74, 69–73.

Centers for Medicare and Medicaid Services (2016), "Quality Payment Program: Delivery System Reform, Medicare Payment Reform, and MACR," Available at `https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/Value-Based-Programs/MACRA-MIPS-and-APMs/MACRA-MIPS-and-APMs.html`.

Chakraborty, B. and Moodie, E. E. M. (2013), *Statistical methods for dynamic treatment regimes: reinforcement learning, causal inference and personalized medicine*, Springer.

Chen, J. H. and Asch, S. M. (2017), "Machine learning and prediction in medicine-beyond the peak of inflated expectations," *N. Engl. J. Med.*, 376, 2507–2509.

Cheng, L., Darnell, G., Chivers, C., Draugelis, M. E., Li, K., and Engelhardt, B. E. (2017), "Sparse multi-output Gaussian processes for medical time series prediction," *ArXiv preprint:1703.09112*.

Cheng-Xian Li, S. and Marlin, B. (2016), "A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification," in *Advances in Neural Information Processing Systems*, p. 18041812.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. (2016a), "RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism," in *Advances in Neural Information Processing Systems*, pp. 3504–3512.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016b), "Using recurrent neural network models for early detection of heart failure onset," *J. Am. Med. Inform. Assoc.*, 24, 361370.

Chow, E. and Saad, Y. (2014), "Preconditioned krylov subspace methods for sampling multivariate gaussian distributions," *SIAM Journal on Scientific Computing*, 36, A588–A608.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015), "A recurrent latent variable model for sequential data," p. 29802988.

Churpek, M. M., Snyder, A., Han, X., Sokol, S., Pettit, N., Howell, M. D., and Edelson, D. P. (2016), "qSOFA, SIRS, and early warning scores for detecting clinical deterioration in infected patients outside the ICU," *Am. J. Respir. Crit. Care Med.*

Cortes-Puch, I. and Hartog, C. S. (2016), "Change is not necessarily progress: revision of the sepsis definition should be based on new scientific insights," *Am. J. Respir. Crit. Care Med.*, 194, 16–18.

Darcy, A. M., Louie, A. K., and Roberts, L. W. (2016), "Machine learning and the profession of medicine," *JAMA*, 315, 551–2.

Dellinger, R. P., Levy, M. M., Rhodes, A., Annane, D., Gerlach, H., Opal, S. M., Sevransky, J. E., Sprung, C. L., Douglas, I. S., Jaeschke, R., Osborn, T. M., Nunnally, M. E., Townsend, S. R., Reinhart, K., Kleinpell, R. M., Angus, D. C.,

Deutschman, C. S., Machado, F. R., Rubenfeld, G. D., Webb, S. A., Beale, R. J., Vincent, J. L., and Moreno, R. (2013), "Surviving sepsis campaign: international guidelines for management of severe sepsis and septic shock: 2012," *Crit. Care Med.*, 39, 165–228.

Desautels, T., Calvert, J., Hoffman, J., Jay, M., Kerem, Y., Shieh, L., Shimabukuro, D., Chettipally, U., Feldman, M. D., Barton, C., Wales, D. J., and Das, R. (2016), "Prediction of sepsis in the intensive care unit with minimal electronic health record data: a machine learning approach," *JMIR Med. Inform.*, 4.

Dezfouli, A. and Bonilla, E. V. (2015), "Scalable inference for Gaussian process models with black-box likelihoods," in *Advances in Neural Information Processing Systems*, pp. 1414–1422.

Doshi-Velez, F. and Kim, B. (2017), "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*.

Duchi, J., Hazan, E., and Singer, Y. (2011), "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, 12, 2121–2158.

Durichen, R., Pimentel, M. A. F., Clifton, L., Schweikart, A., and Clifton, D. A. (2015), "Multitask gaussian processes for multivariate physiological time-series analysis," *IEEE Trans. Biomed. Eng.*, 62, 314–322.

Echouffo-Tcheugui, J. B. and Kengne, A. P. (2012), "Risk models to predict chronic kidney disease and its progression: a systematic review," *PLoS medicine*, 9, e1001344.

Eijkemans, M. J. C., Van Houdenhoven, M., Nguyen, T., Boersma, E., Steyerberg, E. W., and Kazemier, G. (2010), "Predicting the unpredictable: a new prediction model for operating room times using individual characteristics and the surgeon's estimate," *Anesthesiology*, 112, 41–49.

Epstein, L., Dantes, R., Magill, S., and Fiore, A. (2016), "Varying Estimates of Sepsis Mortality Using Death Certificates and Administrative Codes - United States, 1999-2014," *MMWR Morb. Mortal. Wkly. Rep.*, 65, 342–345.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017), "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, 542, 115.

Ferrer, R., Artigas, A., Suarez, D., Palencia, E., Levy, M. M., Arenzana, A., Perez, X. L., Sirvent, J. M., and Edusepsis (2009), "Effectiveness of treatments for severe sepsis: a prospective, multicenter, observational study," *Am. J. Respir. Crit. Care Med.*, 180.

Friedmann, P. D., Breatt, A. S., and Mayo-Smith, M. F. (1996), "Differences in generalists' and cardiologists' perceptions of cardiovascular risk and the outcomes of preventive therapy in cardiovascular disease," *Ann. Intern. Med.*, 124, 414–21.

Futoma, J., Morris, J., and Lucas, J. (2015), "A comparison of models for predicting early hospital readmissions," *J. Biomed. Inform.*, 56, 229–238.

Futoma, J., Sendak, M., Cameron, C. B., and Heller, K. (2016a), "Predicting disease progression with a model for multivariate longitudinal clinical data," in *Machine Learning for Healthcare Conference*, pp. 42–54.

Futoma, J., Sendak, M., Cameron, C. B., and Heller, K. (2016b), "Scalable joint modeling of longitudinal and point process data for disease trajectory prediction and improving management of chronic kidney disease," in *Uncertainty in Artificial Intelligence*, pp. 222–231.

Futoma, J., Hariharan, S., Sendak, M., Brajer, N., Clement, M., Bedoya, A., OBrien, C., and Heller, K. (2017a), "An improved multi-output Gaussian process RNN with real-time validation for early sepsis detection," in *Machine Learning for Healthcare Conference*, pp. 243–254.

Futoma, J., Hariharan, S., and Heller, K. (2017b), "Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier," in *International Conference on Machine Learning*, pp. 1174–1182.

Futoma, J., Lin, A., Sendak, M., Bedoya, A., Clement, M., OBrien, C., and Heller, K. (2017c), "Learning to treat sepsis with multi-output Gaussian process deep recurrent Q-networks," in *NIPS 2017 Workshop on Machine Learning for Health*.

Gage, B. F., Waterman, A. D., Shannon, W., Boechler, M., Rich, M. W., and Radford, M. J. (2001), "Validation of clinical classification schemes for predicting stroke: results from the National Registry of Atrial Fibrillation," *JAMA*, 285, 2864–70.

Gardner-Thorpe, J., Love, N., Wrightson, J., Walsh, S., and Keeling, N. (2006), "The value of Modified Early Warning Score (MEWS) in surgical in-Patients: a prospective observational study," *Ann. R. Coll. Surg. Engl.*, 88, 571–75.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian data analysis*, CRC Press, New York.

Ghahramani, Z. and Beal, M. (2001), "Propagation algorithms for variational Bayesian learning," in *Advances in Neural Information Processing Systems*, p. 507513.

Ghassemi, M., Pimentel, M. A. F., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., and Feng, M. (2015), "A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data," in *Proc. Conf. AAAI Artif. Intell.*, p. 446453.

Goldstein, B. A., Navar, A. M., Pencil, M. J., and Ioannidis, J. P. A. (2017), "Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review," *J. Am. Med. Inform. Assoc.*, 24, 198–208.

González, G., Ash, S. Y., Vegas-Sánchez-Ferrero, G., Onieva Onieva, J., Rahaghi, F. N., Ross, J. C., Díaz, A., San José Estépar, R., and Washko, G. R. (2018), "Disease staging and prognosis in smokers using deep learning in chest computed tomography," *Am. J. Respir. Crit. Care Med.*, 197, 193–203.

Goovaerts, P. (1997), *Geostatistics for natural resources evaluation*, Oxford University Press.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J, K. R., Raman, R., Nelson, P. C., Mega, J. L., and Webster, D. R. (2016), "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, 316, 2402–2410.

Han, J., Slate, E. H., and Pena, E. A. (2007), "Parametric latent class joint model for a longitudinal biomarker and recurrent event," *Stat. Med.*, 26, 5285–5302.

Harutyunyan, H., Khachatrian, H., Kale, D. C., and Galstyan, A. (2017), "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*.

Hausknecht, M. and Stone, P. (2015), "Deep recurrent Q-Learning for partially observable MDPs," in *AAAI Fall Symposium Series*.

Haydar, S., Spanier, M., Weems, P., Wood, S., and Strout, T. (2017), "Comparison of qSOFA score and SIRS criteria as screening mechanisms for emergency department sepsis," *Am. J. Emerg. Med.*, 35, 1730–1733.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. (2017), "Deep reinforcement learning that matters," *arXiv preprint arXiv:1709.06560*.

Henry, J., Pylypchuk, Y., Searcy, T., and Patel, V. (2016), "Adoption of Electronic Health Record Systems among U.S. Non-Federal Acute Care Hospitals: 2008-2015," Available at `https://dashboard.healthit.gov/evaluations/data-briefs/non-federal-acute-care-hospital-ehr-adoption-2008-2015.php`.

Henry, K. E., Hager, D. N., Pronovost, P. J., and Saria, S. (2015), "A targeted real-time early warning score (TREWScore) for septic shock," *Sci. Transl. Med.*, 7, 299ra122.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013), "Gaussian processes for big data," in *Uncertainty in Artificial Intelligence*, pp. 282–290.

Hensman, J., Matthews, A. G., Filippone, M., and Ghahramani, Z. (2015a), "MCMC for variationally sparse Gaussian processes," in *Advances in Neural Information Processing Systems*, pp. 1648–1656.

Hensman, J., Matthews, A. G., and Ghahramani, Z. (2015b), "Scalable variational Gaussian process classification," in *Artificial Intelligence and Statistics*, pp. 351–360.

Hersh, W. R., Weiner, M. G., Embi, P. J., Logan, J. R., Payne, P. R. O., Bernstam, E. V., Lehmann, H. P., Hripcsak, G., Hartzog, T. H., Cimino, J. J., and Saltz, J. H. (2013), "Caveats for the use of operational electronic health record data in comparative effectiveness research," *Med. Care*, 51, S30–S37.

Hinton, G., Srivastava, N., and Swersky, K. (2012), "Lecture 6a: Overview of mini-batch gradient descent," in *Neural Networks for Machine Learning*.

Hochreiter, S. and Schmidhuber, J. (1997), "Long short-term memory," *Neural Computation*, 9, 1735–80.

Hoffman, M., Blei, D., Wang, C., and Paisley, J. (2013), "Stochastic variational inference," *J. Mach. Learn. Res.*, 14, 1303–1347.

Hoiles, W. and van der Schaar, M. (2016), "A Non-parametric learning method for confidently estimating patient's clinical state and dynamics," in *Advances in Neural Information Processing Systems*, p. 20202028.

Holland, P. W. (1986), "Statistics and Causal Inference," *Journal of the American Statistical Association*, 81, 945–960.

Huang, Y. and Hanauer, D. A. (2014), "Patient no-show predictive model development using multiple data sources for an effective overbooking approach," *Appl. Clin. Inform.*, 5, 836–860.

Imbens, G. W. and Rubin, D. B. (2015), *Causal inference in statistics, social, and biomedical sciences*, Cambridge University Press.

Jiang, N. and Li, L. (2016), "Doubly robust off-policy value evaluation for reinforcement learning," p. 652661.

Johnson, A. E. W., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016), "MIMIC-III, a freely accessible critical care database," *Scientific data*, 3, 160035.

Johnson, A. E. W., Pollard, T. J., and Mark, R. G. (2017), "Reproducibility in critical care: a mortality prediction case study," in *Machine Learning for Healthcare Conference*, pp. 361–376.

Johnson, T. L., Rinehart, D. J., Durfee, J., Brewer, D., Batal, H., Blum, J., Oronce, C. I., Melinkovich, P., and Gabow, P. (2015), "For Many Patients Who Use Large Amounts of Health Care Services, the Need Is Intense Yet Temporary," *Health Aff.*, 34, 1312–1319.

Jones, A. E., Shapiro, N. I., Trzeciak, S., Arnold, R. C., Claremont, H. A., Kline, J. A., and EMShockNet (2010), "Lactate clearance vs central venous oxygen saturation as goals of early sepsis therapy: a randomized clinical trial," *JAMA*, 303, 739–46.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999), "An introduction to variational methods for graphical models," *Mach. Learn.*, 37, 183–233.

Kalil, A. C., Johnson, D. W., Lisco, S. J., and Sun, J. (2017), "Early goal-directed therapy for sepsis: a novel solution for discordant survival outcomes in clinical trials," *Crit. Care Med.*, 45, 607–614.

Kawamoto, K., Houlihan, C. A., Balas, E. A., and Lobach, D. F. (2005), "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success," *BMJ*, 330.

Kidney Disease: Improving Global Outcomes CKD-MBD Work Group (2009), "KDIGO clinical practice guideline for the diagnosis, evaluation, prevention, and treatment of Chronic Kidney Disease-Mineral and Bone Disorder (CKD-MBD)." *Kidney Int. Suppl.*, p. S1.

Kim, S., Zeng, D., Chambless, L., and Li, Y. (2012), "Joint models of longitudinal and recurrent events with informative terminal Event," *Stat. Biosci.*, 4, 262–281.

Kingma, D. P. and Ba, J. (2015), "Adam: a method for stochastic optimization," in *International Conference on Learning Representations*.

Kingma, D. P. and Welling, M. (2014), "Auto-encoding variational bayes," in *International Conference on Learning Representations*.

Knaus, W. A., Draper, E. A., Wagner, D. P., and Zimmerman, J. E. (1985), "Apache II: a severity of disease classification system," *Crit. Care Med.*, 13, 818–29.

Koh, P. W. and Liang, P. (2017), "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*, p. 18851894.

Komorowski, M., Gordon, A., Celi, L. A., and Faisal, A. (2016), "A Markov decision process to suggest optimal treatment of severe infections in intensive care," in *NIPS Workshop on Machine Learning for Health*.

Krumholz, H. M. (2014), "Big data and new knowledge in medicine: the thinking, training, and tools needed for a learning health system," *Health Aff.*, 33, 1163–70.

Kumar, A., Roberts, D., Wood, K. E., Light, B., Parrillo, J. E., Sharma, S., Suppes, R., Feinstein, D., Zanotti, S., Taiberg, L., Gurka, D., Kumar, A., and Cheang, M. (2006), "Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock," *Crit. Care Med.*, 34, 1589–96.

Lazarsfeld, P. and Henry, N. (1968), *Latent Structure Analysis*, Houghton Mifflin.

Lei, T., Barzilay, R., and Jaakkola, T. (2016), "Rationalizing neural predictions," in *Conference on Empirical Methods in Natural Language Processing*, pp. 107–117.

Leventhal, R. (2016), "Survey: ACOs still cite lack of interoperability as biggest barrier," Available at `https://www.healthcare-informatics.com/article/survey-acos-still-cite-lack-interoperability-biggest-barrier`.

Levey, A. S., Stevens, L. A., Schmid, C. H., Zhang, Y. L., Castro, A. F., Feldman, H. I., Kusek, J. W., Eggers, P., Van Lente, F., Greene, T., Coresh, J., and CKD-EPI (2009), "A New Equation to Estimate Glomerular Filtration Rate," *Ann. Intern. Med.*, 150, 604–612.

Levy, M. M., Fink, M. P., Marshall, J. C., Abraham, E., Angus, D., Cook, D., Cohen, J., Opal, S. M., Vincent, J. L., Ramsay, G., and International Sepsis Definitions Conference (2003), "2001 SCCM/ESICM/ACCP/ATS/SIS international sepsis definitions conference," *Intensive Care Med.*, 29, 530–538.

Lewis, G. H. (2010a), "Impactibility Models: identifying the subgroup of high-risk patients most amenable to hospital-avoidance programs," *The Milbank Quarterly*, 88, 240–255.

Lewis, R. J. (2010b), "Disassembling goal-directed therapy for sepsis: a first step," *JAMA*, 303, 777–779.

Li, O., Liu, H., Chen, C., and Rudin, C. (2017), "Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions," *arXiv preprint arXiv:1710.04806*.

Lian, W., Henao, R., Rao, V., Lucas, J., and Carin, L. (2015), "A multitask point process predictive mode," in *International Conference on Machine Learning*, pp. 2030–2038.

Lim, W. S., van der Eerden, M. M., Laing, R., Boersma, W. G., Karalus, N., Town, G. I., Lewis, S. A., and Macfarlante, J. T. (2003), "Defining community acquired pneumonia severity on presentation to hospital: an international derivation and validation study," *Thorax*, 58, 377–82.

Lipton, Z. C. (2016), "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015), "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*.

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. (2016a), "Learning to diagnose with LSTM recurrent neural networks," in *International Conference on Learning Representations*.

Lipton, Z. C., Kale, D. C., and R., W. (2016b), "Modeling missing data in clinical time series with RNNs," in *Machine Learning for Healthcare Conference*.

Lipton, Z. C., X., W. Y., and Smola, A. (2017), "Detecting and correcting for label shift with black box predictors," *arXiv preprint arXiv:1802.03916*.

Liu, L. and Huang, X. (2009), "Joint analysis of correlated repeated measures and recurrent events processes in the presence of death, with application to a study on acquired immune deficiency syndrome," *J. Royal Stat. Soc., Series C*, 58, 65–81.

Liu, Y. Y., Li, S., Li, F., Song, L., and Rehg, J. M. (2015), "Efficient learning of continuous-time hidden Markov models for disease progression," in *Advances in Neural Information Processing Systems*, p. 36003608.

Lloyd, C., Gunter, T., Osborne, M. A., and Roberts, S. J. (2015), "Variational inference for Gaussian process modulated Poisson proceses," in *International Conference on Machine Learning*, p. 18141822.

Mbogning, C., Bleakley, K., and Lavielle, M. (2015), "Joint modeling of longitudinal and repeated time-to-event data using nonlinear mixed-effects models and the stochastic approximation expectation-maximization algorithm," *J. Stat. Compute. Simul.*, 85, 1512–1528.

Melis, G., Dyer, C., and Blunsom, P. (2017), "On the state of the art of evaluation in neural language models," *arXiv preprint arXiv:1707.05589*.

Mendelssohn, D. C., Curtis, B., Yeates, K., Langlois, S., MacRae, J. M., Semeniuk, L. M., Camacho, F., McFarlane, P., and STARRT (2011), "Suboptimal initiation of dialysis with and without early referral to a nephrologist," *Nephrol. Dial. Transplant.*, 26, 2859–65.

Mnih, V., Kavukcuoglu, K., Silver, D., , Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015), "Human-level control through deep reinforcement learning," *Nature*, 518, 529–533.

Murphy, K. (2012), *Machine learning: a probabilistic perspective*, MIT Press.

Musoro, J. Z., Geskus, R. B., and Zwinderman, A. H. (2015), "A joint model for repeated events of different types and multiple longitudinal outcomes with application to a follow-up study of patients after kidney transplant," *Biom. J.*, 57, 185–200.

Nemati, S., Ghassemi, M. M., and Clifford, G. D. (2016), "Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach," in *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, p. 29782981.

Nguyen, T. V. and Bonilla, E. V. (2014), "Collaborative Multi-output Gaussian Processes," in *Uncertainty in Artificial Intelligence*, pp. 643–652.

Obermeyer, Z. and Emanuel, E. J. (2016), "Predicting the future: big data, machine learning, and clinical medicine," *N. Engl. J. Med.*, 375, 1216–19.

Obermeyer, Z. and Lee, T. H. (2017), "Lost in thought: the limits of the human mind and the future of medicine," *N. Engl. J. Med.*, 377, 1209–1211.

Olorisade, B. K., Brereton, P., and Andras, P. (2017), "Reproducibility in machine learning-based studies: an example of text mining," .

Pan, S. J. and Yang, Q. (2010), "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, 22, 1345–1359.

Parbhoo, S., Bogojeska, J., Zazzi, M., Roth, V., and Doshi-Velez, F. (2017), "Combining kernel and model based learning for HIV therapy selection," in *AMIA Jt. Summits Trans. Sci. Proc.*, pp. 239–248.

Parikh, R. B., Kakad, M., and Bates, D. W. (2016), "Integrating Predictive Analytics Into High-Value Care: The Dawn of Precision Delivery," *JAMA*, 315, 651–652.

Perotte, A., Ranganath, R., Hirsch, J. S., Blei, D., and Elhadad, N. (2015), "Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis," *J. Am. Med. Inform. Assoc.*, 22, 872880.

Prasad, N., Cheng, L. F., Chivers, C., Draugelis, M., and Engelhardt, B. E. (2017), "A reinforcement learning approach to weaning of mechanical ventilation in intensive care units," in *Uncertainty in Artificial Intelligence*.

PRISM Investigators (2017), "Early, goal-directed therapy for septic shock - a patient-level meta-analysis," *N. Engl. J. Med.*, 376, 2223–2234.

ProCESS Investigators, Yealy, D. M., Kellum, J. A., Huang, D. T., Barnato, A. E., Weissfeld, L. A., Pike, F., Terndrup, T., Wang, H. E., Hou, P. C., LoVecchio, F., Filbin, M. R., Shapiro, N. I., and Angus, D. C. (2014), "A randomized trial of protocol-based care for early septic shock," *N. Engl. J. Med.*, 370, 1683–1693.

Proust-Lima, C. and Taylor, J. M. G. (2009), "Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of posttreatment PSA: a joint modeling approach," *Biostatistics*, 10, 535–549.

Proust-Lima, C., Sene, M., Taylor, J., and Jacqmin-Gadda, H. (2014), "Joint latent class models for longitudinal and time-to-event data: A review," *Stat. Methods Med. Res.*, 23, 74–90.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005), "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, 6, 1939–1959.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009), *Dataset shift in machine learning*, The MIT Press.

Raghu, A., Komorowski, M., Ahmed, I., Celi, L., Szolovits, P., and Ghassemi, M. (2016), "Deep reinforcement learning for sepsis treatment," in *NIPS Workshop on Machine Learning for Health*.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017), "Continuous State-Space Models for Optimal Sepsis Treatment - a Deep Reinforcement Learning Approach," in *Machine Learning for Healthcare Conference*.

Ranganath, R., Gerrish, S., and Blei, D. (2014), "Black box variational inference," in *Artificial Intelligence and Statistics*, p. 814822.

Ranganath, R., Perotte, A. J., Elhadad, N., and Blei, D. M. (2015), "The survival filter: joint survival analysis with a latent time series," in *Uncertainty in Artificial Intelligence*, pp. 742–751.

Rasmussen, C. E. and Williams, C. K. I. (2005), *Gaussian Processes for Machine Learning*, The MIT Press.

Rezende, D., Mohamed, S., and Wierstra, D. (2014), "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning*, p. 12781286.

Rhodes, A., Evans, L. E., Alhazzani, W., Levy, M. M., Antonelli, M., Ferrer, R., Kumar, A., Sevransky, J. E., Sprung, C. L., Nunnally, M. E., Rochwerg, B., Rubenfeld, G. D., Angus, D. C., Annane, D., Beale, R. J., Bellingham, G. J., Bernard, G. R., Chiche, J. D., Coopersmith, C., De Backer, D. P., French, C. J., Fujisjima, S., Gerlach, H., Hidalgo, J. L., Hollenberg, S. M., Jones, A. E., Karnad, D. R., Kleinpell, R. M., Koh, Y., Lisboa, T. C., Machado, F. R., Marini, J. J., Marshall, J. C., Mazuski, J. E., McIntyre, L. A., McLean, A. S., Mehta, S., Moreno, R. P., Myburgh, J., Navalesi, P., Nishida, O., Osborn, T. M., Perner, A., Plunkett, C. M., Ranieri, M., Schorr, C. A., Seckel, M. A., Seymour, C. W., Shieh, L., Shukri, K. A., Simpson, S. Q., Singer, M., Thompson, B. T., Townsend, S. R., Van der Poll, T., Vincent, J. L., Wiersinga, W. J., Zimmerman, J. L., and Dellinger, R. P. (2017), "Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016," *Intensive Care Med.*, 43, 304–377.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016), "Why should i trust you?: explaining the predictions of any classifier," in *Proc. SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, ACM.

Rivers, E., Nguyen, B., Havstad, S., Ressler, J., Muzzin, A., Knoblich, B., Peterson, E., and Tomlanovich, M. (2001), "Early goal-directed therapy in the treatment of severe sepsis and septic shock," *N. Engl. J. Med.*, 345, 1368–1377.

Rizopoulos, D. (2011), "Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data," *Biometrics*, 67, 819–829.

Rizopoulos, D. (2012), *Joint models for longitudinal and time-to-event data: with applications in R*, CRC Biostatistics Series.

Rizopoulos, D., Hatfield, L. A., Carlin, B. P., and Takkenberg, J. J. M. (2014), "Combining dynamic predictions from joint models for longitudinal and time-to-event data using Bayesian model averaging," *J. Am. Stat. Assoc.*, 109, 1385–1397.

Robbins, H. and Monro, S. (1951), "A stochastic approximation method," *Ann. Math. Stat.*, pp. 400–407.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013), "Gaussian processes for time-series modelling," *Phil. Trans. R. Soc. A*, 371, 20110550.

Roski, J., Bo-Linn, G. W., and Andrews, T. A. (2014), "Creating value in health care through big data: opportunities and policy implications," *Health Aff.*, 33, 1115–22.

Ross, A, S., Hughes, M. C., and Doshi-Velez, F. (2017), "Right for the right reasons: training differentiable models by constraining their explanations," in *International Joint Conference on Artificial Intelligence*.

Rothman, M. J., Rothman, S. I., and Beals IV, J. (2013), "Development and validation of a continuous measure of patient condition using the Electronic Medical Record," *J. Biomed. Inform.*, 46, 837–48.

Sackett, D. L., Rosenberg, W. M. C., Gray, J. A. M., Haynes, R. B., and Richardson, W. S. (1996), "Evidence based medicine: what it is and what it isnt," *BMJ*, 312.

Saria, S. and Goldenberg, A. (2015), "Subtyping: What it is and its role in precision medicine," *IEEE Intelligent Systems*, 30, 70–75.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016), "Prioritized experience replay," in *International Conference on Learning Representations*.

Schulam, P. and Saria, S. (2015), "A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure," in *Advances in Neural Information Processing Systems*, p. 748756.

Seymour, C. W., Liu, V. X., Iwashyna, T. J., Brunkhorst, F. M., Rea, T. D., Scherag, A., Rubenfeld, G., Kahn, J. M., Shankar-Hari, M., Singer, M., Deutschman, C. S., Escobar, G. J., and Angus, D. C. (2016), "Assessment of clinical criteria for sepsis: for the third international consensus definitions for sepsis and septic shock (Sepsis-3)," *JAMA*, 315, 762–74.

Seymour, C. W., Gesten, F., Prescott, H. C., Friedrich, M. E., Iwashyna, T. J., Phillips, G. S., Lemeshow, S., Osborn, T., Terry, K. M., and Levy, M. M. (2017), "Time to treatment and mortality during mandated emergency care for sepsis," *N. Engl. J. Med.*, 376, 2235–44.

Shi, J. Q., Murray-Smith, R., and Titterington, D. M. (2005), "Hierarchical Gaussian process mixtures for regression," *Statistics and computing*, 15, 31–41.

Silverman, B. W. (1985), "Some aspects of the spline smoothing approach to non-parametric regression curve fitting," *J. Royal Stat. Soc., Series B*, pp. 1–52.

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J. D., Coopersmith, C. M., Hotchkiss, R. S., Levy, M. M., Marshall, J. C., Martin, G. S., Opal, S. M., Rubenfeld, G. D., van der Poll, T., Vincent, J. L., and Angus, D. C. (2016), "The third international consensus definitions for sepsis and septic shock (Sepsis-3)," *JAMA*, 315, 801–10.

Smart, N., Dieberg, M., Ladhanni, M., and Titus, T. (2008), "Early referral to specialist nephrology services for preventing the progression to end-stage kidney disease," *Cochrane Database of Systematic Reviews*, 18.

Smith, G. B., Prytherch, D. R., Meredith, P., Schmidt, P. E., and Featherstone, P. I. (2013), "The ability of the National Early Warning Score (NEWS) to discriminate patients at risk of early cardiac arrest, unanticipated intensive care unit admission, and death," *Resuscitation*, 84.

Snelson, E. and Ghahramani, Z. (2006), "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, pp. 1257–1264.

Soleimani, H., Hensman, J., and Saria, S. (2017), "Scalable joint models for reliable uncertainty-aware event prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., and Carpenter, J. R. (2009), "Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls," *BMJ*, 338, b2393.

Steyerberg, E. W. (2009), *Clinical prediction models: a practical approach to development, validation, and updating*, Springer, New York.

Sundararajan, M., Taly, A., and Yan, Q. (2017), "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, p. 33193328.

Sutton, R. S. and Barto, A. G. (1998), *Introduction to reinforcement learning*, MIT Press, Cambridge, MA, USA, 1st edn.

Szczech, L. A., Stewart, R. C., Su, H. L., DeLoskey, R. J., Astor, B. C., Fox, C. H., McCullough, P. A., and Vassalotti, J. A. (2014), "Primary care detection of chronic kidney disease in adults with type-2 diabetes: The ADD-CKD Study," *PloS ONE*, 9.

Tangri, N., Stevens, L. A., Griffith, J., Tighiouart, H., Djurdjev, O., Naimark, D., Levin, A., and Levey, A. S. (2011), "A predictive model for progression of chronic kidney disease to kidney failure," *JAMA*, 305, 1553–1559.

Tangri, N., Kitsios, G. D., Inker, L. A., Griffith, J., Naimark, D. M., Walker, S., Rigatto, C., Uhlig, K., Kent, D. M., and Levey, A. S. (2013), "Risk prediction models for patients with chronic kidney disease: a systematic review," *Ann. intern. med.*, 158, 596–603.

Ting, D. S. W., Cheung, C. Y. L., Lim, G., Tan, G. S. W., Quang, N. D., Gan, A., Hamzah, H., Garcia-Franco, R., San Yeo, I. Y., Lee, S. Y., Mun Wong, E. Y., Sabanayagam, C., Baskaran, M., Ibrahim, F., Tan, N. C., Finkelstein, E. A., Lamoureux, E. L., Wong, I. Y., Bressler, N. M., Sivaprasad, S., Varma, R., Jonas, J. B., He, M. G., Cheng, C. Y., Ming Cheung, G. C., Aung, T., Hsy, W., Lee, M. L., and Wong, T. Y. (2017), "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *JAMA*, 318, 2211–2223.

Titsias, M. (2009), "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, pp. 567–574.

Torio, C. M. and Moore, B. J. (2016), "National Inpatient Hospital Costs: The Most Expensive Conditions by Pay, 2013," *Healthcare Cost and Utilization Project (HCUP) Statistical Briefs*, 204.

Tuot, D., Plantinga, L., Hsu, C., Jordan, R., Burrows, N. R., Hedgeman, E., Yee, J., Saran, R., Powe, N. R., and CDC-CKD (2011), "Chronic kidney disease awareness among individuals with clinical markers of kidney dysfunction," *Clin. J. Am. Soc. Nephrol.*, 6, 1838–1844.

van Hasselt, H., Guez, A., and Silver, D. (2016), "Deep reinforcement learning with double Q-Learning," in *Proc. Conf. AAAI Artif. Intell.*, p. 20942100.

Vincent, J. L., Moreno, R., Takala, J., Willatts, S., De Mendonca, A., Bruining, H., Reinhart, C. K., Super, P. M., and Thijs, L. G. (1996), "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure," *Intensive Care Med.*, 22, 707–10.

Wackernagel, H. (1998), *Multivariate Geostatistics: An Introduction with Applications*, Springer-Verlag, 2nd edition edn.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. (2016), "Dueling network architectures for deep reinforcement learning," in *International Conference on Machine Learning*, p. 19952003.

Wasson, J. H., Sox, H. C., Neff, R. K., and Goldman, L. (1985), "Clinical prediction rules - applications and methodological standards," *N. Engl. J. Med.*, 313, 793–9.

Watkins, C. J. C. H. and Dayan, P. (1992), "Q-learning," *Mach. Learn.*, 8, 279–292.

Weil, A. R. (2014), "Big data in health: a new era for research and patient care," *Health Aff.*, 33, 1110.

Weiskopf, N. G., Rusanov, A., and Weng, C. (2013), "Sick patients have more data: the non-random completeness of electronic health records," in *American Medical Informatics Association Annual Symposium Proceedings*, p. 1472.

Wilson, P. W. F., DAgostino, R. B., Levy, D., Belanger, A. M., Silbershatz, H., and Kannel, W. B. (1998), "Prediction of coronary heart disease using risk factor categories," *Circulation*, 97, 1837–47.

Xing, Z., Jian, P., and Philip, S. Y. (2012), "Early classification on time series," *Knowledge and information systems*, 31, 105–127.

Yoon, J., Alaa, A. M., Hu, S., and van der Schaar, M. (2016), "ForecastICU: a prognostic decision support system for timely prediction of intensive care unit admission," in *International Conference on Machine Learning*, p. 16801689.

Zhengping, C., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2016), "Recurrent neural networks for multivariate time series with missing values," *arXiv preprint arXiv:1606.01865*.

Biography

Joseph David Futoma was born on July 11, 1992 in Bethesda, MD. He graduated summa cum laude from Dartmouth College with High Honors in 2013 with an A.B. in Mathematics and was a member of Phi Beta Kappa. He graduated from Duke University with a Ph.D. in Statistical Science in 2018, earning an M.S. in Statistical Science in 2016. He was a First Year Statistical Science Research Fellow, a Statistical & Applied Mathematical Sciences Institute (SAMSI) Research Fellow, and a National Defense Science and Engineering Graduate (NDSEG) Research Fellow. He was also a member of a winning team in the LinkedIn Economic Graph Challenge. He will start a postdoc fellowship in the summer of 2018 at Harvard University in the Center for Research on Computation and Society, advised by Finale Doshi-Velez.