Journal of Machine Learning Research 19 (2018) 1-62 Submitted 12/16; Revised 11/17; Published 8/18

A Hidden Absorbing Semi-Markov Model for Informatively Censored Temporal Data: Learning and Inference

Ahmed M. Alaa[†] AHMEDMALAA@UCLA.EDU
[†]*Electrical Engineering Department*
University of California, Los Angeles (UCLA)
Los Angeles, CA 90095-1594, USA

Mihaela van der Schaar[*,†] MIHAELA.VANDERSCHAAR@ENG.OX.AC.UK
[*]*Department of Engineering Science*
University of Oxford
Parks Road, Oxford OX1 3PJ, UK

Editor: Edoardo M. Airoldi

Abstract

Modeling continuous-time physiological processes that manifest a patient's evolving clinical states is a key step in approaching many problems in healthcare. In this paper, we develop the *Hidden Absorbing Semi-Markov Model* (HASMM): a versatile probabilistic model that is capable of capturing the modern electronic health record (EHR) data. Unlike existing models, the HASMM accommodates irregularly sampled, temporally correlated, and informatively censored physiological data, and can describe non-stationary clinical state transitions. Learning the HASMM parameters from the EHR data is achieved via a novel *forward-filtering backward-sampling* Monte-Carlo EM algorithm that exploits the knowledge of the end-point clinical outcomes (informative censoring) in the EHR data, and implements the E-step by sequentially sampling the patients' clinical states in the reverse-time direction while conditioning on the future states. Real-time inferences are drawn via a forward-filtering algorithm that operates on a virtually constructed discrete-time *embedded Markov chain* that mirrors the patient's continuous-time state trajectory. We demonstrate the prognostic utility of the HASMM in a critical care prognosis setting using a real-world dataset for patients admitted to the Ronald Reagan UCLA Medical Center. In particular, we show that using HASMMs, a patient's clinical deterioration can be predicted 8-9 hours prior to intensive care unit admission, with a 22% AUC gain compared to the Rothman index, which is the state-of-the-art critical care risk scoring technology.

Keywords: Hidden Semi-Markov Models, Medical Informatics, Monte Carlo methods.

1. Introduction

Modeling the clinical conditions of a patient using evidential physiological data is a ubiquitous problem that arises in many healthcare settings, including disease progression modeling (Schulam and Saria (2015); Mould (2012); Wang et al. (2014); Jackson et al. (2003); Sweeting et al. (2010); Liu et al. (2015)) and critical care prognosis (Moreno et al. (2005); Matos et al. (2006); Yoon et al. (2016); Hoiles and van der Schaar (2016); Alaa et al. (2016)). Accurate physiological modeling in these settings confers an *instrumental value* that manifests

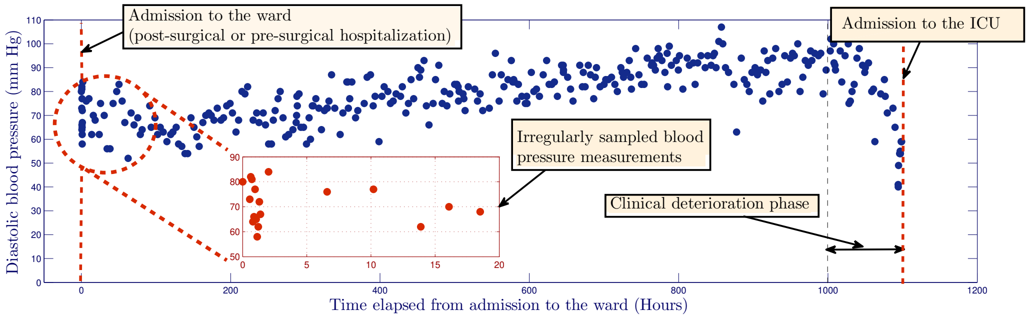


Figure 1: An episode of the diastolic blood pressure measurements (as recorded in the EHR) for a patient hospitalized in a regular ward for 50 days and then admitted to the ICU after the ward staff realized she is clinically deteriorating.

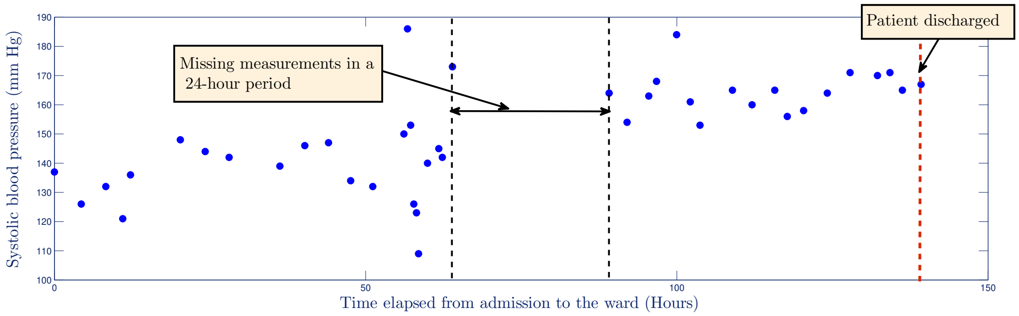


Figure 2: An episode of the systolic blood pressure measurements for a patient hospitalized in a regular ward for 6 days and then discharged home by the ward staff. Measurements are missing in a 24-hour period during the patient's stay in the ward.

in the ability to provide early diagnosis, individualized treatments and timely interventions (e.g. early warning systems in hospital wards (Yoon et al. (2016)), early diagnosis for Scleroderma patients (Varga et al. (2012); Alaa and van der Schaar (2016)), early detection of a progressing breast cancer (Bartkova et al. (2005)), etc). Physiological modeling also confers an *epistemic value* that manifests in the knowledge extracted from data about the progression and severity phases of a disease (Stelfox et al. (2012))), or the short-term dynamics of the physiological behavior of critically ill patients (Li-wei et al. (2013)).

The recent availability of data in the electronic health records (EHR)[1] creates a promising horizon for establishing rich and complex physiological models (Gunter and Terry (2005)). Modern EHRs comprise *episodic* data records for individual (anonymized) patients; every patient's episode is a temporal sequence of clinical findings (e.g. visual field index for Glaucoma patients (Liu et al. (2015)), CD4 cell counts for HIV-infected patients (Guihenneuc-Jouyaux et al. (2000)), etc), lab test results (e.g. white cell blood count for

[1] A recent data brief from the Office for National Coordinator (ONC) for healthcare technology shows that the adoption of EHR in US hospitals exhibited a spectacular increase from 9.4% in 2008, 27.6% in 2011, to 75.5% in 2014 (Charles et al. (2015)).

post-operative patients under immunosuppressive drugs (Cholette et al. (2012)), etc), or vital signs (e.g. blood pressure and O_2 saturation (Yoon et al. (2016))). The time span of these episodes may be as short as few days in short-term hospitalization episodes (e.g. patients with solid tumors, hematological malignancies or neutropenia who are hospitalized in regular wards before or after a surgery (Kause et al. (2004); Hogan et al. (2012); Kirkland et al. (2013))), or as long as few years in longitudinal episodes (e.g. chronic obstructive pulmonary disease may evolve from a mild Stage I to a very severe Stage IV over a time span of 10 years (Pedersen et al. (2011); Wang et al. (2014))). **In this paper, we develop a versatile time-series model that provides means for accurate real-time risk prognostication of adverse clinical outcomes.** Other applications of the model include but are not limited to modeling default data in quantitative finance (Giampieri et al. (2005)), and fault detection in general dynamic systems (Smyth (1994)). In the next Subsection, we expose our modeling rationale and challenges posed by the structure of modern EHR data. We conclude this Section by summarizing our contributions in Subsection 1.2.

1.1 Modeling Rationale and Challenges

1.1.1 Rationale

Previous physiological models have branched into two different modeling paths with respect to the way a patient's clinical states are defined. One strand of literature adopts *fully observable models*; these models assume that clinical states are quantifiable via *observable* clinical markers or disease severity measures (e.g. PFVC in Scleroderma (Schulam and Saria (2015)), GFR in kidney disease (Eddy and Neilson (2006)), etc). Another strand of literature adopts *latent variable models*, which assume that clinical states are latent and manifest only through proximal, noisy physiological measurements. Table 1 lists some notable previous works that fall under each modeling category[2].

Table 1: Modeling methodologies in previous works.

Methodology	Previous Works
Fully observable models	• HIV (Dessie (2014); Foucher et al. (2005)) • Chronic kidney diseases (Eddy and Neilson (2006)) • Scleroderma (Schulam and Saria (2015)) • ICU (Ghassemi et al. (2015)).
Latent variable models	• Alzheimer (Chen and Zhou (2011)) • HIV (Guihenneuc-Jouyaux et al. (2000)) • Glaucoma (Liu et al. (2015)) • Comorbidities (Wang et al. (2014)).

[2] While the models in (Schulam and Saria (2015)) and (Ghassemi et al. (2015)) involve latent variables that designate patient subtypes, the clinical states in both works are considered to be captured via observable bio-markers (PFVC in the former and the Cerebrovascular Autoregulation index in the latter).

Our modeling choice is to go with a latent variable model for the following reasons:

- In a wide range of problems, a concrete clinical marker that can be directly used as a surrogate for the patient's true clinical condition is **not** available. This is especially true in critical care settings where no solid definition or measure of a "clinical state" exists (Li-wei et al. (2013)). Previous works that adopted a clinical risk score as a surrogate for the clinical state in critical care settings have found that other physiological features, when augmented with the clinical risk score, still hold a significant predictive power with respect to end-point clinical outcomes (Ghassemi et al. (2015)). This implies that a clinical risk score or a severity of illness measure (such as APACHE II, SAPS and SOFA (Knaus et al. (1991); Subbe et al. (2001))) is not a sufficient measure of a patient's true clinical condition, and hence cannot be reliably modeled as an observable clinical state.

- The same line of argument extends to disease progression models: (Jackson et al. (2003)) has shown that significant modeling gain can be attained by treating clinical markers and diagnostic assessments as noisy manifest variables for the patient's true clinical state rather than defining a clinical state in terms of those markers.

- For various chronic disease, such as HIV, Scleroderma, and kidney disease, progression stages are well defined in terms of observable clinical markers (CD4 cell count, PFVC and GFR). However, a latent variable model can help validate and assess the current domain knowledge-based clinical practice guidelines by learning alternative, data-driven guidelines. Other diseases, such as COPD, have their progression stages manifesting only through symptoms (e.g. chronic bronchitis, emphysema and chronic airway obstruction (Wang et al. (2014))), which may or may not accurately reflect the disease's true state, and hence a latent variable model is necessary.

- Conclusive clinical markers that reveal a patient's true state may be available only occasionally in a patient's longitudinal episode. For instance, in a breast cancer progression setting, most of the data points associated with a patient's longitudinal episode would be imaging test results (e.g. BI-RADS scores of a mammogram or an MRI (Gail and Mai (2010); Taghipour et al. (2013))), which are noisy markers for the existence of a tumor, whereas a conclusive biopsy result that truly reveals whether the patient is in a preclinical or clinical breast cancer state may not be available because the patient did not undergo a biopsy test.

- A fully observable model does not provide diagnostic utility since it assumes that an already observable clinical marker provides an immediate, domain-knowledge-based diagnosis for the patient. Contrarily, a latent variable model leaves room for diagnoses to be learned from evidential data by learning the association between physiological evidence and clinical states, which may help inform and improve clinical practice.

1.1.2 Challenges

Hidden Markov Models (HMMs) and their variants have been widely deployed as temporal latent variable models for dynamical systems (Smyth (1994); Zhang et al. (2001); Giampieri et al. (2005); Genon-Catalot et al. (2000); Ghahramani and Jordan (1997)). Such models have achieved considerable success in various applications, such as topic modeling (Gruber et al. (2007)), speaker diarization (Fox et al. (2011b)), and speech recognition (Rabiner (1989)). However, the nature of the clinical setting, together with the format of the modern EHR data pose the following set of serious challenges that confound classical HMM models:

(A) Non-stationarity: Recently developed disease progression models, such those in (Wang et al. (2014)) and (Liu et al. (2015)), use conventional stationary Markov chain models. In particular, they assume that state transition probabilities are independent of time. However, this assumption is seriously at odds with even casual observational studies which show that the probability of transiting from the current state to another state depends on the time spent in the current state (Lagakos et al. (1978); Huzurbazar (2004); Gillaizeau et al. (2015)). This effect, which violates the memorylessness assumptions adopted by continuous-time Markovian models, was verified in patients who underwent renal transplantation (Foucher et al. (2007, 2008)), patients who are HIV infected (Joly and Commenges (1999); Dessie (2014); Foucher et al. (2005)), and patients with chronic obstructive pulmonary disease (Bakal et al. (2014)).

(B) Irregularly spaced observations: The times at which the clinical findings of a patient (vital signs or lab tests) are observed is controlled either by clinicians (in the case of hospitalized inpatients), or by the patient's visit times (in the case of a chronic disease follow up). The time interval between every two measurements may vary from one patient to another, and may also vary for the same patient within her episode. This is reflected in the structure of the episodes in the EHR records, as shown in Figure 1 and 2. Figure 1 depicts an actual diastolic blood pressure episode for a patient hospitalized in a regular ward for 1200 hours (50 days)[3]. The patient's stay in the ward was concluded with an admission to the ICU after the ward staff realized she was clinically deteriorating. As we can see, the blood pressure measurements in the first 20 hours were initially taken with a rate of 1 sample per hour, and then later the rate changed to 1 sample every 5 hours[4]. While some recent works have argued for the parametrization of time in longitudinal data via the natural event sequence (Hripcsak et al. (2015)), it is often the case that the sampling times are themselves informative of the patients' clinical well-being (Alaa et al. (2017)). Thus, a direct application of a regular, discrete-time HMM (e.g. the models in (Murphy (2002); Fox et al. (2011b,a); Rabiner (1989); Yu (2010); Matos et al. (2006); Guihenneuc-Jouyaux et al. (2000))) will not suffice for jointly describing the latent states and observations, and hence ensuring accurate inferences.

(C) Discrete observations of a continuous-time phenomena: A patient's physiological signals and latent states evolve in continuous time; however, the observed physiological measurements are gathered at discrete time steps that can differ from one physiological signal to another. (One alternative view of such a structure is to think of a time series with irregularly sampled multidimensional measurements and with missing data in every measurement (Lipton et al. (2016)). We do not address data that is missing **not** at random

[3] A detailed description for the data involved in this paper is provided in Section 5.

[4] While Figure 1 illustrates a short-term episode for a critical care patient, similar effects are experienced in longitudinal episodes for patients with chronic disease (see Figure 4 in (Wang et al. (2014))).

in this paper.) The intervals between observed measurements can vary quite significantly; as we can see in Figure 2, the systolic blood pressure for a patient who stayed in a ward for 140 hours exhibits an entire day without measurements [5]. This means that the patient may encounter multiple hidden state transitions without any associated observed data. These effects make learning and inference problems more complicated since the inference algorithms need to consider potential unobserved trajectories of state evolution between every two timestamps. This challenge has been recently addressed in (Nodelman et al. (2012); Wang et al. (2014); Liu et al. (2015)), but only on the basis of memoryless Markov chain models for the hidden states, for which tractable inferences that rely on the solutions to Chapman-Kolmogorov equations can be executed. However, incorporating non-stationarity in state transitions (i.e. addressing challenge (A)) would make the problem of reasoning about a continuous-time process through discrete observations much more complicated.

(D) Lack of supervision: The episodes in the EHR may be labeled with the aid of domain knowledge (e.g. the stages and symptoms of some chronic diseases, such as chronic kidney disease (Eddy and Neilson (2006)), are known to clinicians and may be provided in the EHR). However, in many cases, including the case of (post or pre-operative) critical care, we do not have access to any labels for the patients' states. Hence, unsupervised learning approaches need to be used for learning model parameters from EHR episodes. While unsupervised learning of discrete-time HMMs has been extensively studied and is well understood (e.g. the Baum-Welch EM algorithm is predominant in such settings (Zhang et al. (2001); Yu (2010); Rabiner (1989))), the problem of unsupervised learning of continuous-time models for which both the patient's states and state transition times are hidden is far less understood, and indeed far more complicated.

(E) Censored observations: Episodes in the EHR are usually terminated by an informative intervention or event, such as death, ICU admission, discharge, etc. This is known as *informative censoring* (Scharfstein and Robins (2002); Huang and Wolfe (2002); Link (1989)). Unlike classical HMM settings where training sets comprise fixed length, or arbitrarily-censored, HMM sequence instances, a typical EHR dataset would comprise a set of episodes with different durations, and the duration of each episodes is itself informative of the state trajectory. Learning in such settings requires algorithms that can efficiently compute the likelihood of observing a set of episodes conditioned on their durations and terminating states.

1.2 Summary of Contributions

In order to address the challenges above, we develop a new model –which we call the *Hidden Absorbing Semi-Markov Model* (HASMM)– as a versatile generative model for a patient's (physiological) episode as recorded in the EHR. The HASMM captures non-stationary transitions for a patient's clinical state via a continuous-time semi-Markov model with explicitly specified state sojourn time distributions. Informative censoring is captured via absorbing states that designate clinical endpoint outcomes (e.g. cardiac arrest, mortality, recovery,

[5] This may have resulted due to the patient undergoing a surgery or an intervention, or because the EHR recording system accidentally did not receive the data from the clinicians during that day.

etc); entering an absorbing state of an HASMM stimulates censoring events (e.g. clinical deterioration leads to an ICU admission which terminates the physiological observations for a monitored patient in a ward, etc). Observable variables are modeled via a multi-task Gaussian process (Bonilla et al. (2007)), for which the observation times (i.e. follow up visits, vital sign gathering, lab tests, etc) are modeled as a point process. Using multi-task Gaussian process with state-dependent hyper-parameters, an HASMM accounts for both correlations among different physiological variables, in addition to the temporal correlations among the observation variables that are generated by the same hidden state during its sojourn period. In that sense, an HASMM is a segment model (Ostendorf et al. (1996)) and also a *state-switching* model (Fox et al. (2011a))).

To allow for real-time inference of a patient's state, we develop a forward-filtering HASMM inference algorithm that can estimate a patient's latent state using her history of irregularly sampled physiological measurements. The inference algorithm operates by constructing a virtual, discrete-time *embedded Markov chain* that fully describes the patient's state transitions at observation times. The embedded Markov chain is constructed in an offline stage by solving a system of *Volterra integral equations of the second kind* using the *successive approximation* method; the solution to this system of equations, which parallels the Chapman-Kolmogorov equations in ordinary Markov chains, describe the HASMM's semi-Markovian state transitions as observed at arbitrarily selected discrete timestamps.

Offline learning of the HASMM model parameters from patients' episodes in an EHR is a daunting task. Since the HASMM is a continuous-time model, we cannot directly use the classical Baum-Welch EM algorithms for learning its parameters (Rabiner (1989)). Moreover, the semi-Markovianity of an HASMM yields an intractable integral in the E-step of the Expectation-Maximization (EM) formulation. Since the HASMM's state transitions are not captured by the conventional continuous-time Markov chain transition rate matrices, we cannot make use of the *Expm* and *Unif* methods that were used in (Hobolth and Jensen (2011)), and more recently in (Liu et al. (2015)) for evaluating the integrals involved in the E-step of learning continuous-time HMMs. To address this challenge, we develop a novel *forward-filtering backward-sampling Monte Carlo EM* (FFBS-MCEM) algorithm that approximates the integral involved in the E-step by efficiently sampling the latent clinical trajectories conditioned on observations in the EHR by exploiting the informative censoring of the patients' episodes. The FFBS-MCEM algorithm samples the latent clinical states of every (offline) patient episode in the EHR as follows: it starts from the known clinical endpoints, and sequentially samples the patient's states by traversing in the reverse-time direction while conditioning on the future states, and then uses the sampled state trajectories to evaluate a Monte Carlo approximation for the E-step.

2. The Hidden Absorbing Semi-Markov Model (HASMM)

2.1 Abstract Model

We start by describing the HASMM's hidden state evolution process, and then we describe the structure of its observable variables.

2.1.1 Hidden States

We consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{R}_+}, \mathbb{P})$, over which a continuous-time stochastic process $X(t)$ is defined on $t \in \mathbb{R}_+$. The process $X(t)$ corresponds to a temporal trajectory of the patient's hidden clinical states, which take on values from a finite *state-space* $\mathcal{X} = \{1, 2, \ldots, N\}$. Because the process $X(t)$ takes on only finitely many values, it can be decomposed in the form[6]

$$X(t) = \sum_n X_n \cdot \mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}, \tag{1}$$

where $(X(t))_{t\in\mathbb{R}_+}$ is a càdlàg path, $X_n \in \mathcal{X}$, and the interval $[\tau_n, \tau_{n+1})$ is the time interval accommodating the n^{th} hidden state. Every path $(X(t))_{t\in\mathbb{R}_+}$ on the stochastic basis $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{R}_+}, \mathbb{P})$ is a *semi-Markov path* (Janssen and De Dominicis (1984); Durrett (2010)), where the *sojourn time* of state n, which we denote as $S_n = \tau_{n+1} - \tau_n$, is drawn from a *state-specific* distribution $v_j(S_n = s \,|\lambda_j) = d\mathbb{P}(S_n = s \,|X_n = j)$, with λ_j being a state-specific *duration parameter* associated with state $j \in \mathcal{X}$. Unlike ordinary time-homogeneous semi-Markov transitions, in which the transition probabilities among states are assumed to be constant conditioned on there being a transition from the current state (Gillaizeau et al. (2015); Murphy (2002); Johnson and Willsky (2013); Yu (2010); Dewar et al. (2012); Guédon (2007)), our model accounts for *duration-dependent* semi-Markov transitions, i.e. we have that

$$\mathbb{P}(X_{n+1} = j | X_n = i, S_n = s) = g_{ij}(s), \tag{2}$$

where $g_{ij} : \mathbb{R}_+ \to [0, 1]$, $\forall i, j \in \mathcal{X}$ is a *transition function* for which $\frac{\partial g_{ij}(s)}{\partial s}$ is well defined, and $\sum_{j=1}^{N} g_{ij}(s) = 1, \forall s \in \mathbb{R}_+, i \in \mathcal{X}$.

Now consider the bi-variate (renewal) process $(X_n, S_n)_{n\in\mathbb{N}_+}$, which comprises the sequence of states and sojourn times. Semi-Markovianity of $X(t)$ implies that $(X_n, S_n)_{n\in\mathbb{N}_+}$ satisfies the following condition on its transition probabilities

$$\begin{aligned}\mathbb{P}(X_{n+1} = j, S_n \leq s \,|\, \mathcal{F}_{\tau_n}) &= \mathbb{P}(X_{n+1} = j, S_n \leq s \,|\, X_n = i)\\ &= \mathbb{P}(X_{n+1} = j \,|\, X_n = i, S_n \leq s) \cdot \mathbb{P}(S_n \leq s \,|\, X_n = i)\\ &= \mathbb{E}_{S_n}\left[g_{ij}(S_n) \,|\, S_n \leq s\right] \cdot V_i(s \,|\, \lambda_i) = \bar{g}_{ij}(s) \cdot V_i(s \,|\, \lambda_i),\end{aligned} \tag{3}$$

where $\{X_n = i\} \in \mathcal{F}_{\tau_n}$, $V_i(.)$ is the cumulative distribution function of state i's sojourn time, and $\bar{g}_{ij}(s)$ is the probability mass function that reflects the probability that a patient's next state being j given that she was at state i and her sojourn time in i is less than (or equal to) s. Based on (3), we define the *semi-Markov transition kernel* as a matrix-valued function $\mathbf{Q} : \mathbb{R}_+ \to [0, 1]^{N\times N}$ that describes the dynamics of $X(t)$ in continuous time, with entries $\mathbf{Q}(s) = (Q_{ij}(s))_{i,j\in\mathcal{X}}$ that are given by

$$Q_{ij}(s) = \bar{g}_{ij}(s) \cdot V_i(s \,|\, \lambda_i). \tag{4}$$

The initial state X_1 is random[7], and the initial state distribution is given by $\mathbf{p}^o = [\,p_1^o, \ldots, p_N^o\,]^T$, where $p_j^o = \mathbb{P}(X(0) = j)$, and $\sum_{j=1}^{N} p_j^o = 1$.

[6] By convention, we set $\tau_1 = 0$.

[7] We do not consider left-censored observations in this model.

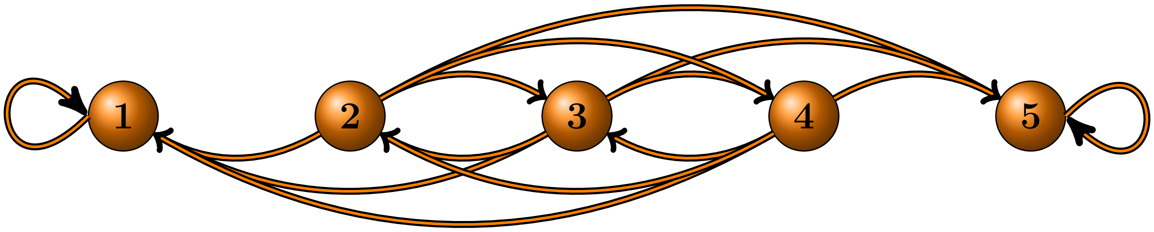


Figure 3: The Markov chain model for a 5-state HASMM.

We assume that whenever the patient enters either state 1 or state N, she remains there forever[8]. Therefore, we model states $\{1, N\}$ as *absorbing states*, whereas we model the remaining states in $\mathcal{X} \setminus \{1, N\}$ as *transient states* that represent intermediate levels of risk. We define and interpret states 1 and N as follows:

- **State** 1 is denoted as the *safe state*, and represents the state at which the patient is at minimum (or no) risk (e.g. clinically stable post-operative patient, etc).
- **State** N is denoted as the *catastrophic state*, and represents the state at which the patient is at severe risk or encounters an adverse event (e.g. a very severe stage of a chronic disease (Bakal et al. (2014)), a cardiac or respiratory arrest (Subbe et al. (2001)), mortality (Knaus et al. (1991)), etc).

We do not assume that the transient states are ordered linearly in terms of clinical risk. Following the assumptions in (Murphy (2002); Johnson and Willsky (2013)), we eliminate the self-transitions for all transient states by setting $g_{ii}(s) = 0, Q_{ii}(s) = 0, \forall s \in \mathbb{R}_+, i \in \mathcal{X} \setminus \{1, N\}$, whereas we restrict the transitions from states 1 and N to self-transitions only, i.e. $g_{ii}(s) = 1, i \in \{1, N\}$. Figure 3 depict the Markov chain for the sequence $\{X_n\}_{n \in \mathbb{N}_+}$.

We define $\mathcal{A}_1$ as the event that the path $(X(t))_{t \in \mathbb{R}_+}$ is absorbed in the safe state 1, i.e. $\mathcal{A}_1 = \{\lim_{t \to \infty} X(t) = 1\}$, and $\mathcal{A}_N$ as the event that $(X(t))_{t \in \mathbb{R}_+}$ is absorbed in the catastrophic state N, i.e. $\mathcal{A}_N = \{\lim_{t \to \infty} X(t) = N\}$. Since $(X(t))_{t \in \mathbb{R}_+}$ is an absorbing semi-Markov chain[9], we know that $\mathbb{P}(\mathcal{A}_1 \vee \mathcal{A}_N) = 1$, and since the events $\mathcal{A}_1$ and $\mathcal{A}_N$ are mutually exclusive, it follows that $\mathbb{P}(\mathcal{A}_N) = 1 - \mathbb{P}(\mathcal{A}_1)$. The quantity $\mathbb{P}(\mathcal{A}_N)$ describes a patient's prior risk of ending in the catastrophic state, whereas $\mathbb{P}(\mathcal{A}_N \,|\mathcal{F}_t)$ describes the patient's posterior risk of ending in the catastrophic state having observed its evolution history up to time t[10]. Define T_s as an $\mathcal{F}$-stopping time representing the absorption time

[8] The model can be easily extended to accommodate an arbitrary number of competing absorbing states.

[9] We assume that the transition functions $g_{i1}(s)$ and $g_{iN}(s)$ for any transient state i is non-zero for every s. Hence, it follows that $(X(t))_{t \in \mathbb{R}_+}$ is an absorbing semi-Markov chain since it has 2 absorbing states, each of which can be visited starting from any other state (Durrett (2010)).

[10] In the clinical applications under consideration, transient states can be ordered by their respective relative risks of encountering event $\mathcal{A}_N$ in the subsequent transitions, i.e. in a 5-state chain, it is more likely for the patient to be absorbed in state 5 in the future when it is in state 4 than when it is in state 3. For instance, it is more likely for a patient's chronic obstructive pulmonary disease that is currently assessed to have a severity degree of GOLD1 (mild severity as defined in the GOLD standard Pedersen et al.

of the path $(X(t))_{t \in \mathbb{R}_+}$ in either state 1 or state N, i.e.

$$T_s = \inf\{t \in \mathbb{R}_+ : X(t) \in \{1, N\}\}.$$

Finally, we define K as the (random) number of state realizations in the sequence $\{X_n\}_{n=1}^{K}$ up to the stopping time T_s, which has to be concluded by either state 1 or N, e.g. when $|\mathcal{X}| = 4$, the sequences $\{1\}, \{4\}, \{2, 3, 2, 2, 3, 4\}$, and $\{3, 2, 1\}$ are valid, random-length realizations of $\{X_n\}_{n=1}^{K}$, and each represents a certain state evolution trajectory for the patient.

2.1.2 Observations and Censoring

The path $(X(t))_{t \in \mathbb{R}_+}$ is unobservable; what is observable is a corresponding process $(Y(t))_{t \in \mathbb{R}_+}$ on $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in \mathbb{R}_+}, \mathbb{P})$, the values of which are drawn from an *observation-space* $\mathcal{Y}$, and whose distributional properties are dependent on the latent states' path $(X(t))_{t \in \mathbb{R}_+}$. The observable process $(Y(t))_{t \in \mathbb{R}_+}$ can be put in the form

$$Y(t) = \sum_{n} Y_n(t) \cdot \mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}, \tag{5}$$

where $(Y(t))_{t \in \mathbb{R}_+}$ is a càdlàg path, comprising a sequence of function-valued variables $\{Y_n(t)\}_{n=1}^{K}$, with $Y_n : [\tau_n, \tau_{n+1}) \to \mathcal{Y}$. Even though the path $(Y(t))_{t \in \mathbb{R}_+}$ is accessible, only a sequence of irregularly spaced samples of it is observed over time, and is denoted by $\{Y(t_m)\}_{t_m \in \mathcal{T}}$, where $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$ is the set of observed measurements, and M is the total number of such measurements. We say that the process is censored if $M < \infty$; typical episodes in an EHR are censored: observations stop at some point of time due to a release from care, an ICU admission, mortality, etc.

The sampling times in $\mathcal{T}$ represent the times at which a patient with a chronic disease took clinical tests (i.e. time intervals in $\mathcal{T}$ spans years), or the times at which clinicians have gathered vital signs for a monitored critically ill patient in a hospital ward (i.e. time intervals in $\mathcal{T}$ span days or hours). We assume that the sampling times in $\mathcal{T}$ are drawn from a *point-process* $\Phi = \sum_{m \in \mathbb{N}_+} \delta_{t_m}$, with δ_t being the Dirac measure. The point-process Φ is assumed to be independent of the latent states path[11]. Define $\mathcal{T}_n$ as the set of M_n samples that are gathered during the interval[12] $[\tau_n, \tau_{n+1})$, i.e. $\mathcal{T}_n = \{t_m : t_m \in \mathcal{T}, t_m \in [\tau_n, \tau_{n+1})\}$, $M_n = |\mathcal{T}_n|$, and $\sum_n M_n = M$. Since $\mathcal{T}_n$ could possibly be empty ($\mathcal{T}_n = \emptyset$), some states can have no corresponding observations (i.e. an inpatient may exhibit a transition to a deteriorating state during the night, even though her blood pressure were not measured during the night.

(2011)) to progress (in the near future) to a severity degree of GOLD2 (moderate) rather than GOLD3 (severe).

[11] This means that the sampling times are uninformative of the latent states; which simplifies the inference problem. The HASMM model can be extended to incorporate a state-dependent sampling process using a Cox process (Lando (1998)) or a Hawkes process (Hawkes and Oakes (1974)) to modulate the point-process intensity;however, such an extension would result in a significantly harder inference problem. A good discussion on conditional intensity models can be found in (Qin and Shelton (2015)).

[12] Note that what is observed is a sequence of sampling times $\mathcal{T}$, the elements of which are not labeled by the corresponding state indexes, for that the states are latent, i.e. the sets $\mathcal{T}_n$ are unobserved partitions of $\mathcal{T}$.

Recall the illustration in Figure 2).

The paths $\{Y_n(t)\}_{n=1}^{K}$ are assumed to be conditionally independent given the hidden sequence of states $\{X_n\}_{n=1}^{K}$, and hence we have that

$$\{Y(t_m)\}_{t_m \in \mathcal{T}_n} \perp\!\!\!\perp \{Y(t_m)\}_{t_m \in \mathcal{T}_{n+1}} \,|\, X_n, X_{n+1}\,, \forall n \in \{1, 2, \ldots, K-1\}.$$

The observed samples generated under every state X_n and sampled at the times in $\mathcal{T}_n$ are drawn from $\mathcal{Y}$ according to a distribution $\mathbb{P}(\{Y(t_m)\}_{t_m \in \mathcal{T}_n} | X_n = j, \Theta_j)$, where Θ_j is an *emission parameter* that controls the distributional properties of the observations generated under state j.

The number of observation samples is finite: the observed sequence is *censored* at some point of time, which we call the censoring time T_c, after which no more observation samples are available. Censoring reflects an external intervention/event that terminated the observation sequence, i.e. death, intensive care unit (ICU) admission, etc. Censoring is *informative* (Scharfstein and Robins (2002);Huang and Wolfe (2002);Link (1989)), because the censoring time is correlated with the absorption time T_s, and T_s strictly precedes T_c (in an almost sure sense). That is, T_c is an $\mathcal{F}$-stopping time that is given by $T_c = T_s + S_K$, i.e. once the patient enters state 1 or state N, the observations stop after the patient's sojourn time in that state (i.e. observations stop after a time S_K from the entrance in the absorbing state). Therefore, the duration distributions $v_1(s|\lambda_1)$ and $v_N(s|\lambda_N)$ of states 1 and N are used to determine the censoring times conditioned on the chain $\{X_n\}_{n=1}^{K}$ being absorbed at time T_s.

Every sample from the HASMM is an episode comprising a random-length sequence of hidden states $\{X_n\}_{n=1}^{K}$, and a random-length sequence of observations $\{Y(t_m)\}_{m=1}^{M}$ together with the associated observation times. We only observe $\{Y(t_m)\}_{m=1}^{M}$; the path of latent states $X(t)$, the number of realized states K, the association between observations and states (i.e. the sets $\mathcal{T}_n$) are all unobserved, which makes the inference problem very challenging, but captures the realistic EHR data format and the associated inferential hurdles. In the next subsection, we specify the model's generative process and present an algorithm to generate episodic samples from an HASMM.

2.2 Model Specification and Generative Process

As have been discussed in Subsection 2.1, the hidden and observables variables of an HASMM can be listed as follows:

- **Hidden variables:** The hidden states sequence $\{X_n\}_{n=1}^{K}$ and the states' sojourn times $\{S_n\}_{n=1}^{K}$ (or equivalently, the transition times $\{\tau_n\}_{n=1}^{K}$).

- **Observable variables:** The observed episode $\{Y(t_m)\}_{m=1}^{M}$ and the associated sampling times $\mathcal{T} = \{t_m\}_{m=1}^{M}$.

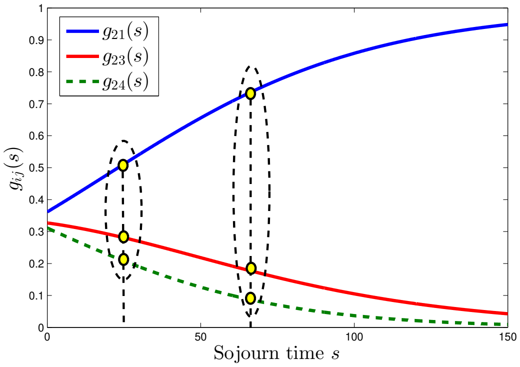


Figure 4: Exemplary transition functions $(g_{2j})_{j=1}^{4}$ for a 4-state HASMM.

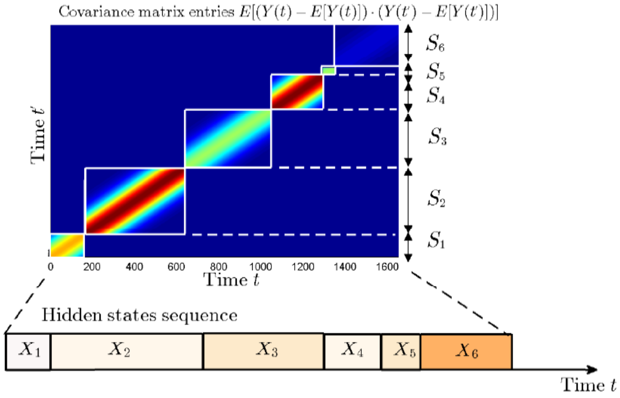


Figure 5: Depiction for the correlation structure of the observable variables for an underlying state sequence $\{X_n\}_{n=1}^{6}$.

The HASMM model parameters that generate both the hidden and observable variables are encompassed in the parameter set Γ, i.e.

$$\Gamma = \Bigg(\underbrace{N}_{\text{State cardinality}}, \underbrace{\lambda = \{\lambda_j\}_{j=1}^{N}}_{\text{State duration}}, \underbrace{\mathbf{p}^o}_{\text{Initial states}}, \underbrace{\mathbf{Q} = \{Q_{ij}(s)\}_{i,j=1}^{N}}_{\text{Transitions}}, \underbrace{\mathbf{\Theta} = \{\Theta_j\}_{j=1}^{N}}_{\text{Emission}} \Bigg).$$

The total number of parameters in an HASMM is $N^2 + 3N(E+1)$.

2.2.1 Distributional specifications for the hidden variables

We model the state sojourn time of each state $i \in \mathcal{X}$ via a Gamma distribution. The selection of a Gamma distribution ensures that the generative process encompasses ordinary continuous-time Markov models for the path $(X(t))_{t \in \mathbb{R}_+}$, since the exponential distribution[13] is a special case of the Gamma distribution (Durrett (2010)). Thus, if the underlying physiology of the patient is naturally characterized by memoryless state transitions, this will be automatically learned from the data via the parameters of the Gamma distribution. The sojourn time distribution for state i is given by

$$v_i(s|\lambda_i = \{\lambda_{i,s}, \lambda_{i,r}\}) = \frac{1}{\Gamma(\lambda_{i,s})} \cdot \lambda_{i,r}^{\lambda_{i,s}} \cdot s^{\lambda_{i,s}-1} \cdot e^{-s \cdot \lambda_{i,r}}, s \geq 0,$$

where $\lambda_{i,s} > 0$ and $\lambda_{i,r} > 0$ are the shape and rate parameters of the Gamma distribution respectively.

Now we specify the structure of the transition kernel $\mathbf{Q}(s) = (Q_{ij}(s))_{i,j}, i, j \in \mathcal{X}$. Recall from (4) that the each element in the transition kernel matrix can be written as $\mathbb{E}_S\left[g_{ij}(S)|S \leq s\right] \cdot V_i(s|\lambda_i)$. Having specified the distribution $v_i(s|\lambda_i)$ as a Gamma distribution, it remains to specify the function $g_{ij}(s)$ in order to construct the elements of $\mathbf{Q}(s)$.

[13] Note that a semi-Markov chain reduces to a Markov chain if the sojourn times are exponentially distributed.

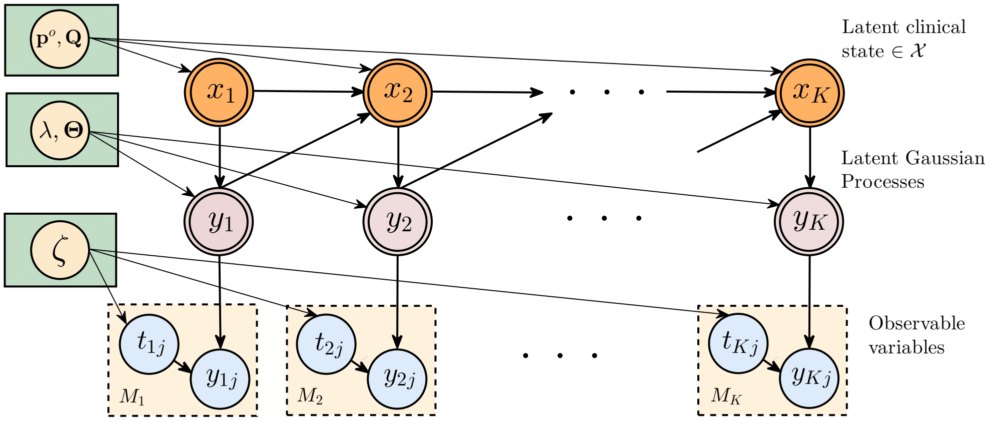


Figure 6: A basic graphical model for the HASMM. The arrow between the (function-valued) variable y_n and the latent state X_{n+1} designates the dependence of state transitions on the sojourn time of the previous state (duration-dependence).

The transition functions $(g_{ij}(s))_{i,j}$ are given by *multinomial logistic* functions as follows

$$\begin{aligned} g_{ij}(s) &= \frac{e^{(\eta_{ij}+\beta_{ij}\cdot s)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}\cdot s)}}, \forall i \neq j, i \notin \{1, N\} \\ g_{ii}(s) &= 0, \forall i \in \{2, \ldots, N-1\}, \\ g_{ii}(s) &= 1, \forall i \in \{1, N\}, \end{aligned} \tag{6}$$

where $\eta_{ij}, \beta_{ij} \in \mathbb{R}_+$. The parameters $(\eta_{ij})_{j=1}^{N}$ determine the baseline values for the transition probability mass from state i to state j, i.e. $g_{ij}(0)$, whereas the parameters β_{ij} controls the dependence of the transition probability mass on the sojourn time[14]. If $\beta_{ij} = 0$, then we have that $g_{ij}(s) = g_{ij}(0) = \frac{e^{\eta_{ij}}}{\sum_{k=1}^{N} e^{\eta_{ik}}}, \forall s \in \mathbb{R}_+$, i.e. the transition probability out of state i remains constant irrespective of the sojourn time in that state. In the limit when s goes to infinity, β_{ij} dominates the functional form in (6), and we have that $g_{ij}(\infty) = \arg\max_j \beta_{ij}$. Figure 4 depicts exemplary transition functions for a 4-state HASMM.

[14] Similar effects for the sojourn time on the transition probabilities has been demonstrated in the progression of breast cancer from healthy to preclinical states in (Taghipour et al. (2013)), where age (the main risk factor for breast cancer) was shown to affect the probability of progressing across the states of healthy to preclinical, clinical and death. These effects may be also prevailing in other diseases, or in critical care settings where the length of time during which a patient stays clinically stable may imply that the patient is more likely to transit to a more healthy state in the future. Through the HASMM model, we can recognize whether or not this effect is evident in the EHR data, i.e. whether the transition function reflects an underlying homogeneous (if $g_{ij}(s)$ is independent of s) or duration-dependent transitions by learning the parameter β_{ij}. Moreover, the parameter β_{ij} is defined per state; the HASMM model can capture scenarios where transitions are duration-independent from some states, but are duration-dependent from others.

2.2.2 Distributional specifications for the observable variables

As explained in Subsection 2.1, the observable process $Y(t)$ can be decomposed as $Y(t) = \sum_{n=1}^{K} Y_n(t) \cdot \mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}$, where the paths $(Y_n(t))_{n=1}^{K}$ are conditionally independent given the state sequence $\{X_n\}_{n=1}^{K}$. Since observations are drawn from $Y(t)$ at arbitrarily, and irregularly spaced time instances $\mathcal{T}$, we have to model the distributional properties of $Y(t)$ in continuous time. We model every path $Y_n(t)$ defined over $[\tau_n, \tau_{n+1})$ as a segment drawn from a multi-task Gaussian Process (GP), with a hyper-parameter set Θ_i that depends on the corresponding latent state $X_n = i$ (Rasmussen (2006); Bonilla et al. (2007)). The input to the multi-task GP is the time variable and the output is the set of physiological variables at a certain point of time. The GP associated with every state $X_n = i$ is parametrized by a constant mean function $m_i(t) = m_i$, a *squared-exponential* covariance kernel $k_i(t, t^{'}) = \sigma_i^2 \, e^{-\frac{1}{2\ell_i^2}\,||t-t^{'}||^2}$, and a "free-form" covariance matrix Σ_i between the different physiological measurements (Bonilla et al. (2007)). Thus, for a E-dimensional physiological stream $Y(t) = (Y^1(t), \ldots, Y^E(t))$, the observations for state i are generated as follows

$$\left\langle Y_i^l(t) \cdot Y_i^v(t^{'}) \right\rangle = \Sigma_i(l, v) \, \cdot \, k_i(t, t^{'}), \qquad \{Y_i^l(t)\}_{t \in \mathcal{T}, 1 \leq l \leq E} \sim \mathcal{N}(m_i(t), \mathbf{\Sigma}_i),$$

where $\mathbf{\Sigma}_i(l, v, t, t^{'}) = \left\langle Y_i^l(t) \cdot Y_i^v(t^{'}) \right\rangle$. The GP hyper-parameters associated with state i are given by $\Theta_i = (m_i, \sigma_i, \Sigma_i, \ell_i)$, i.e. $Y_n(t)|X_n = i \sim \mathcal{GP}(\Theta_i)$.

We note that the HASMM model is a *segment model* (Ostendorf et al. (1996); Murphy (2002); Yu (2010); Guédon (2007)), i.e. observation samples that are defined within the sojourn time of the same state are correlated, but observation samples in different states are independent. **The segmental nature of the model allows for easily handling irregular sampling of temporally correlated observation at the cost of introducing discontinuities of the observed data at the state transition times; in all clinical settings of interest, capturing temporal correlations of irregular observations is crucial whereas the continuity of observations is of less relevance.** The model can also be viewed as a state-switching model, but for which the transition dynamics do not need to be linear as in (Georgatzis et al. (2016); Fox et al. (2011a)), but rather depend on the covariance kernel $k_i(t, t^{'})$. Figure 5 depicts the correlation structure of the observable variables in terms of the covariance matrix of a discrete version of $Y(t)$ generated under a specific hidden state sequence. We can see that conditioned on the hidden state sequence, the covariance matrix is a block diagonal matrix, where the sizes of the blocks are random and are determined by the states' sojourn times.

Figure. 6 depicts the graphical model for an HASMM. In Figure. 6, the variables y_n are function-valued and correspond to the finite-duration, continuous-time functions $\{Y_n(t)\}_{t \in \mathcal{T}_n}$. **The arrow between the (function-valued) y_n and the latent state x_{n+1} designates the dependence of the transition probabilities on the state sojourn time (i.e. the domain over which y_n is non-zero).** In Appendix A, we present an algorithm (`GenerateHASMM(Γ)`) for sampling episodes from an HASMM with a hyper-parameter set Γ; Figure 7 depicts an exemplary episode sampled via Algorithm 7.

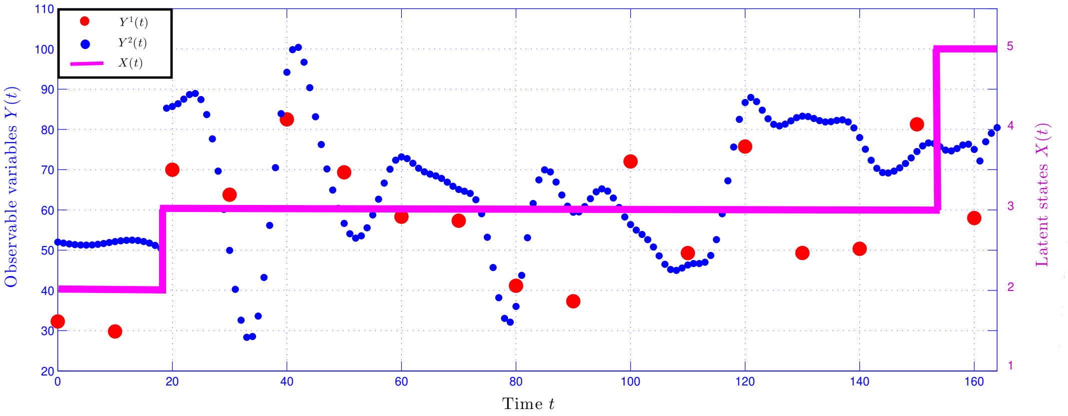


Figure 7: An episode generated by `GenerateHASMM(Γ)` with $N = 5$. The realized hidden state sequence (upper) is $\{2, 3, 5\}$, and is absorbed in state 5. The physiological stream $(Y^1(t), Y^2(t))$ is 2-dimensional and stream $Y^2(t)$ is sampled more intensely than $Y^1(t)$.

3. Inference in Hidden Absorbing Semi-Markov Models

In this Section, we develop an online algorithm that carries out diagnostic and prognostic inferences for a monitored patient's episode in real-time. Given an ongoing realization of an episode $\{y(t_1), y(t_2), \ldots, y(t_m)\}$ at time t_m (before the censoring time T_c), and the HASMM model parameter Γ that has generated this realization, we aim at carrying out the following inference tasks:

- **Diagnosis:** Infer the patient's current clinical state, i.e. compute

$$\mathbb{P}(X(t_m) = j \,|\, Y(t_1) = y(t_1), \ldots, Y(t_m) = y(t_m), \Gamma), \; \forall j \in \mathcal{X}.$$

- **Prognostic Risk Scoring:** Compute the patient's risk of absorption in the catastrophic state, i.e.

$$\mathbb{P}(\mathcal{A}_N \,|\, Y(t_1) = y(t_1), \ldots, Y(t_m) = y(t_m), \Gamma).$$

In the rest of this Section, we drop the conditioning on Γ for notational brevity. The first inference task corresponds to disease severity estimation for patients with chronic disease, or clinical acuity assessment for critical care patients. The second task corresponds to risk scoring for future adverse events for patients who have been monitored for some period of time, i.e. the risk of developing a future preclinical or clinical breast cancer state (Gail and Mai (2010)), the risk of clinical deterioration for post-operative patients in wards (Rothman et al. (2013)), the risk of mortality for ICU patients (Knaus et al. (1985)), etc.

3.1 Challenges facing the HASMM Inference Tasks

The inference tasks discussed in the previous Subsection are confronted with 3 main challenges –listed hereunder– that hinder the direct deployment of classical forward-backward message-passing routines.

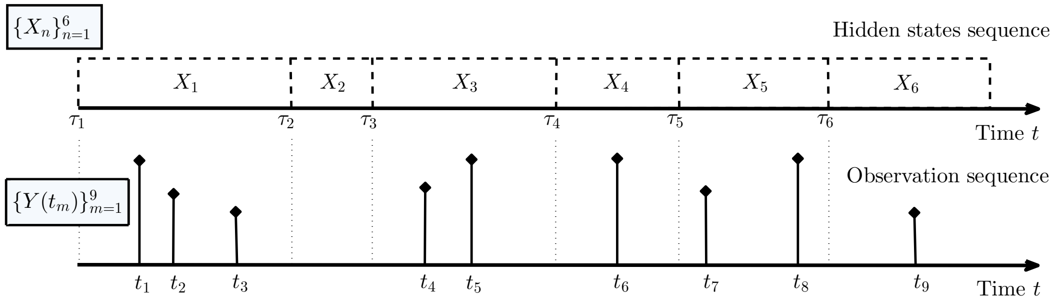


Figure 8: An exemplary HASMM episode with 6 hidden state realizations and 9 observed samples.

1. In addition to the clinical states $\{X_n\}_{n=1}^{K}$ being unobserved, the transition times among the states, $\{\tau_n\}_{n=1}^{K}$, are also unobserved (i.e. we do not know the time at which the patient's state changed). Thus, unlike the discrete-time models in (Murphy (2002); Johnson and Willsky (2013); Yu (2010); Dewar et al. (2012); Guédon (2007)), in which we know that the underlying states switch sequentially in a (known) one-to-one correspondence with the observations, in an HASMM the association between states and observations is unknown. Figure 8 depicts an exemplary HASMM episode with 6 realized states and 9 observations samples; in this realization, the association between the observations $\{Y(t_1), Y(t_2), Y(t_3)\}$ and state X_1 is hidden. The importance of reasoning about the hidden transition times is magnified by the duration-dependence of the transition probabilities that govern the sequence $\{X_n\}_{n=1}^{K}$.

2. Since observations are made at random and arbitrary time instances, some transitions may not be associated with any evidential data. That is, as it is the case for state X_2 in Figure 8, there is no guarantee that for every state X_n, an observation is drawn during its occupancy, i.e. $[\tau_n, \tau_{n+1})$. In a practical setting, the inference algorithm should be able to reason about the state trajectories even in silence periods that come with no observations (recall the example in Figure 2 where observations of a critical care patient's systolic blood pressure stop for an entire day). Hence, one cannot directly discretize the time variable and use the discrete-time HMM inference algorithms (e.g. the algorithms in (Rabiner (1989))) since in that case we would exhibit time steps that come with no associated observations, and with potential state transitions.

3. The HASMM model assumes that observations that belong to the same state are correlated (e.g. in Figure 8, each of the subset of observations $\{Y(t_1), Y(t_2), Y(t_3)\}$, $\{Y(t_4), Y(t_5)\}$ and $\{Y(t_7), Y(t_8)\}$ are not drawn independently conditioned on the latent state since they are sampled from a GP), thus we cannot use the variable-duration and explicit-duration HSMM inference algorithms in (Murphy (2002); Johnson and Willsky (2013); Yu (2010); Guédon (2007)), as those assume that all observations are conditionally independent given the latent states. Our model is closer to a segment-HSMM model (Yu (2010); Guédon (2007)), but with irregularly spaced observations and an underlying duration-dependent state evolution process, which requires a different construction of the forward messages.

In the following Subsection, we develop a forward filtering algorithm that deals with episodes generated from an HASMM and addresses the above challenges.

3.2 The HASMM Forward Filtering Algorithm

Given a realization of an episode $\{y(t_1), y(t_2), \ldots, y(t_m)\}$ at time t_m, the posterior probability of the patient's current clinical state $X(t_m)$ is given by[15]

$$\begin{aligned} \mathbb{P}(X(t_m) = j \,|\, y(t_1), \ldots, y(t_m), \mathcal{T}) &= \frac{d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}{d\mathbb{P}(y(t_1), \ldots, y(t_m) | \mathcal{T})} \\ &= \frac{d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}{\sum_{j'=1}^{N} d\mathbb{P}(X(t_m) = j', y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}. \end{aligned} \tag{7}$$

The above application of Bayes' rule implies that, given the observation times $\mathcal{T}$, computing the joint probability density $d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})$ suffices for computing the posterior probability of the patient's clinical states. As it is the case for the conventional HMM setting, we denote these joint probabilities as the *forward messages* $\alpha_m(j \,|\, \mathcal{T}) = d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})$.

Since the HASMM is a segment model, the conventional notion of the forward messages $\alpha_m(j \,|\, \mathcal{T})$ does not suffice for constructing the forward filtering algorithm since we need to account for the latent correlation structures between the (conditionally-dependent) observations (Murphy (2002)). To that end, we define $\alpha_m(j, w \,|\, \mathcal{T})$ as the forward message for the j^{th} state at the m^{th} observation time (i.e. t_m) *with a lag* w as follows

$$\alpha_m(j, w \,|\, \mathcal{T}) = d\mathbb{P}(X(t_m) = j, \{t_u\}_{u=m-w+1}^{m} \in \mathcal{T}_n, t_{m-w} \in \mathcal{T}_{n'}, \{y(t_u)\}_{u=1}^{m} \,|\, \mathcal{T}), \tag{8}$$

for some $n, n^{'} \in \mathbb{N}_+$, and $n \neq n^{'}$. That is, the forward message $\alpha_m(j, w \,|\, \mathcal{T})$ is simply the joint probability that the current state is j, that the associated observations are $(y(t_1), \ldots, y(t_m))$, and that the current state has lasted for the last w measurements. For notational brevity, denote the event $\left\{\{t_u\}_{u=m-w+1}^{m} \in \mathcal{T}_n, t_{m-w} \in \mathcal{T}_{n'}\right\}$ as $\psi(m, w)$. Thus, $\alpha_m(j, w \,|\, \mathcal{T})$ can be written as

$$\alpha_m(j, w \,|\, \mathcal{T}) = \sum_{i=1}^{N} \sum_{w^{'}=1}^{m-w} d\mathbb{P}(X(t_m) = j, \psi(m, w), X(t_{m-w}) = i, \psi(m-w, w^{'}), \{y(t_u)\}_{u=1}^{m} \,|\, \mathcal{T}),$$

which can be decomposed using the conditional independence properties of the states, observable variables and sojourn times as follows

$$d\mathbb{P}(X(t_m) = j, \psi(m, w), X(t_{m-w}) = i, \psi(m-w, w^{'}), \{y(t_u)\}_{u=1}^{m}) =$$

$$d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j, \psi(m, w)) \times \underbrace{\mathbb{P}(X(t_m) = j \,|\, X(t_{m-w}) = i, \psi(m-w, w^{'}))}_{p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'}))} \times$$

$$\underbrace{\mathbb{P}(\psi(m, w) \,|\, X(t_m) = j)}_{V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j)} \times \underbrace{d\mathbb{P}(X(t_{m-w}) = i, \psi(m-w, w^{'}), \{y(t_u)\}_{u=1}^{m-w})}_{\alpha_{m-w}(i, w^{'})}, \tag{9}$$

[15] We use the notation $d\mathbb{P}$ to denote a probability density defined with respect to $(\Omega, \mathcal{F}, \mathbb{P})$.

where we have dropped the conditioning on $\mathcal{T}$ for notational brevity. The first term, $d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j, \psi(m,w))$, is the probability density of the observable variables in $\{y(t_u)\}_{u=m-w+1}^{m}$ conditioned on the hidden state being $X(t_m) = j$ and that the time instances $\{t_u\}_{u=m-w+1}^{m}$ reside in the sojourn time of $X(t_m) = j$. The second term, $p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'}))$, is the *interval transition probability*, i.e. the probability that the state sequence transits to state j after a period $t_m - t_{m-w}$, given that its sojourn time in state $X(t_{m-w}) = i$ at t_m is at least $t_m - t_{m-w+1}$, and at most $t_m - t_{m-w-w^{'}}$. The third term is the probability that the sojourn time in state $X(t_m) = j$ is between $t_m - t_{m-w+1}$ and $t_m - t_{m-w}$, whereas the fourth term, $\alpha_{m-w}(i, w^{'})$, is the $(m-w)^{th}$ forward message with a lag of $w^{'}$. Thus, we can write the m^{th} forward message with a lag w as follows

$$\alpha_m(j,w) = d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j) \times$$

$$\sum_{i=1}^{N} \sum_{w^{'}=1}^{m-w} p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'})) \cdot (V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j)) \cdot \alpha_{m-w}(i, w^{'}). \tag{10}$$

As we can see in (10), one can express $\alpha_m(j,w)$ using a recursive formula that makes use of the older forward messages $\{\alpha_{m-w}(i, w^{'})\}_{w=1}^{m}$, where $\alpha_o(i, w^{'}) = 0$, which allows for an efficient dynamic programming algorithm to infer the patient's clinical state in real-time.

The construction of the forward messages in (10) parallels the structure of forward message-passing in segment-HSMM (See Section 1.2 in (Murphy (2002)) and Section 4.2.2 in (Yu (2010))), but with the following differences. In (10), the time interval between every two observation samples is irregular, which reflects in the correlation between the observations in $\{y(t_u)\}_{u=m-w+1}^{m}$ (depends on the covariance kernel of the GP, and the probability of the current latent state's sojourn time being encompassing the most recent w samples, i.e. $(V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j))$. However, the most challenging ingredient of the forward message is the interval transition probability $p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'}))$. This is because unlike the discrete-time HSMM models in (Murphy (2002); Yu (2010)), which exhibit transitions only at discrete time steps that are always accompanied with evidential observations, i.e. no hidden transitions can occur between observation samples, and the transitions among hidden states are duration-independent, in an HASMM, transitions can occur at arbitrary time instances, multiple transitions can occur between two observation samples, and transitions are duration-dependent.

In order to evaluate the term $p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'}))$, we construct a virtual (discrete-time) trivariate *embedded Markov chain* $\{X(t_m), t_{m-w}, t_{m-w+1}\}$, the transition probabilities of which are equal to the interval transition probabilities. In the recent work in (Liu et al. (2015)), a similar embedded Markov chain analysis was conducted for a CT-HMM (Continuous-time HMM), but for which the underlying state evolution process was assumed to be a duration-independent ordinary Markov chain for which the expressions for the interval transition probabilities are readily available by virtue of the exponential distributions of the memoryless state sojourn times.

Recall from Subsection 2.1.1 that the semi-Markov kernel of the hidden state sequence $\{X_n\}_{n=1}^{K}$ is defined as $Q_{ij}(\tau) = \mathbb{P}(X_{n+1} = j, S_n \leq \tau | X_n = i)$, i.e. the probability that the sequence transits from state i to state j given that the sojourn time in i is less than or equal to τ. Theorem 1 establishes the methodology for computing the interval transition probabilities $p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'}))$ using the parameters of an HASMM. In Theorem 1, we define $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ as a matrix-valued function $\tilde{\mathbf{P}} : \mathcal{S} \to [0,1]^{N \times N}$, $\mathcal{S} = \{(\tau, \underline{s}, \bar{s}) : \tau \in \mathbb{R}_+, \bar{s} \in \mathbb{R}_+, \underline{s} \leq \bar{s}\}$, the entries of which are given by

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \underbrace{\left[\begin{array}{c|c|c|c} \tilde{p}_{11}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{21}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{N1}(\tau, \underline{s}, \bar{s}) \\ \tilde{p}_{12}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{22}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{N2}(\tau, \underline{s}, \bar{s}) \\ \vdots & \vdots & & \vdots \\ \tilde{p}_{1N}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{2N}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{NN}(\tau, \underline{s}, \bar{s}) \end{array}\right]}_{\text{Size } N \times N \text{ matrix}}.$$

In addition, we define a *truncated semi-Markov kernel* as

$$\bar{Q}_{ij}(\tau, \underline{s}, \bar{s}) = \int_{s=\underline{s}}^{\bar{s}} \left(\bar{g}_{ij}(\tau + s) - \bar{g}_{ij}(s)\right) \cdot \frac{V_i(\tau + s|\lambda_i) - V_i(s|\lambda_i)}{1 - V_i(s|\lambda_i)} \cdot dV_i(s|\lambda_i),$$

a scalar-valued function $\bar{Q}_i(\tau, \underline{s}, \bar{s}) = \sum_{j \in \mathcal{X} \setminus \{i\}} \bar{Q}_{ij}(\tau, \underline{s}, \bar{s})$, and a matrix-valued function

$$\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s}) = \underbrace{\left[\begin{array}{c|c|c|c} 0 & \bar{Q}_{21}(\tau, \underline{s}, \bar{s}) & \cdots & \bar{Q}_{N1}(\tau, \underline{s}, \bar{s}) \\ \bar{Q}_{12}(\tau, \underline{s}, \bar{s}) & 0 & \cdots & \bar{Q}_{N2}(\tau, \underline{s}, \bar{s}) \\ \vdots & \vdots & & \vdots \\ \bar{Q}_{1N}(\tau, \underline{s}, \bar{s}) & \bar{Q}_{2N}(\tau, \underline{s}, \bar{s}) & \cdots & 0 \end{array}\right]}_{\text{Size } N \times N \text{ matrix}}.$$

Theorem 1 (Interval transition probabilities) *Let $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ be the solution to the following integral equation*

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathbf{I}_{N \times N} - \operatorname{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \int_{u=0}^{\tau} \frac{\partial \bar{\mathbf{Q}}(u, \underline{s}, \bar{s})}{\partial u} \times \tilde{\mathbf{P}}(\tau - u, 0, 0)\, du, \tag{11}$$

for the three independent variables $(\tau, \underline{s}, \bar{s}) \in \mathcal{S}$. Then, the interval transition probability p_{ij} is given by $p_{ij}(t_m - t_{m-w}, \psi(m-w, w^{'})) = \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}), \forall i, j \in \mathcal{X}$, at $\tau = t_m - t_{m-w}$, $\underline{s} = t_m - t_{m-w+1}$, and $\bar{s} = t_m - t_{m-w+w'}$.

Proof *See Appendix B.* ■

Theorem 1 follows from a *first-step analysis* that is akin to the derivation of the conventional Chapman-Kolmogorov equations in ordinary Markov chains (Kulkarni (1996)). The integral equation in (11) is a (matrix-valued) non-homogeneous *Volterra integral equation of the second kind* (Polyanin and Manzhirov (2008)). It can be easily demonstrated that a closed-form solution that hinges on conventional kernel methods cannot be obtained. Hence, we resort to a numerical method in order to solve (11) for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$, $\forall(\tau, \underline{s}, \bar{s}) \in \mathcal{S}$. Before presenting the numerical method, we reformulate (11) as follows

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathbf{I}_{N\times N} - \text{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \left(\frac{\partial \bar{\mathbf{Q}}(., \underline{s}, \bar{s})}{\partial u} \star \tilde{\mathbf{P}}(., 0, 0)\right)(\tau), \quad (12)$$

where $\star$ is an element-wise convolution operator. (12) follows from (11) by the fact that the integral in (11) is a convolution integral; (12) can be expressed as follows

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})), \quad (13)$$

where the (functional) operator $\mathcal{B}\{\mathbf{Q}\}(\tilde{\mathbf{P}})$ is given by

$$\begin{aligned}&\mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})) = \\ &\mathbf{I}_{N\times N} - \text{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \mathscr{F}^{-1}\left\{\mathscr{F}\left\{\frac{\partial \bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})}{\partial \tau}\right\} \cdot \mathscr{F}\left\{\tilde{\mathbf{P}}(\tau, 0, 0)\right\}\right\},\end{aligned} \quad (14)$$

where $\mathscr{F}$ is the Fourier transform operator, and the transforms in (14) are all taken with respect to τ.

The solution to (13) can be obtained via the *successive approximation* method (Opial (1967)) as follows. We initialize the function $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ with the truncated semi-Markov kernel[16] $\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})$, and then iteratively apply the operator $\mathcal{B}(.)$ to obtain a new value for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ until convergence. That is, the successive approximation procedure goes as follows

$$\begin{aligned}&\tilde{\mathbf{P}}^{o}(\tau, \underline{s}, \bar{s}) = \bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s}) \\ &\text{While } \left\| \tilde{\mathbf{P}}^{z}(\tau, \underline{s}, \bar{s}) - \tilde{\mathbf{P}}^{z-1}(\tau, \underline{s}, \bar{s}) \right\|_{\infty} > \epsilon \\ &\tilde{\mathbf{P}}^{z}(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}^{z-1}(\tau, \underline{s}, \bar{s})).\end{aligned} \quad (15)$$

The following Theorem establishes the validity of the procedure in (15) as a solver for (13). Before presenting the statement of Theorem 2, we define the function space $\mathcal{P}$ as follows

$$\mathcal{P} = \left\{\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) : \; \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}) \in [0, 1], \sum_{j} \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}) = 1, \tilde{p}_{ij}(0, \underline{s}, \bar{s}) = \delta_{ij}, (\tau, \underline{s}, \bar{s}) \in \mathcal{S}\right\},$$

where δ_{ij} is the Kronecker delta function.

Theorem 2 (Convergence of successive approximations) *The functional $\mathcal{B}\{\bar{\mathbf{Q}}\}(\tilde{\mathbf{P}})$ has a unique fixed-point $\tilde{\mathbf{P}}^{*}$ in $\mathcal{P}$, and the successive approximation procedure in (15) always converges to the fixed point, i.e. $\tilde{\mathbf{P}}^{\infty}(\tau, \underline{s}, \bar{s}) = \tilde{\mathbf{P}}^{*}(\tau, \underline{s}, \bar{s})$, starting from any initial value $\tilde{\mathbf{P}}^{o}(\tau, \underline{s}, \bar{s}) \in \mathcal{P}$.*

Proof *See Appendix C.* ■

[16] This is a reasonable initialization since the entries of the semi-Markov kernel correspond to interval transition probabilities conditioned on there being no intermediate transitions on the way from state i to state j.

Algorithm 1 Constructing a look-up table of interval transition probabilities

```
procedure TransitionLookUp($\Gamma$, $\epsilon$)
    Input: HASMM parameters $\Gamma$ and precision $\epsilon$
    Output: A look-up table $\left[\tilde{p}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})\right]_{i,j,a,b,c}$
    Set the values of $A, B$ and $C$ (number of steps), $\Delta\tau$ (step sizes)
    for $a = 1$ to $A$, $b = 1$ to $B$, $c = 1$ to $C$ do
        $g^{\tau}_{ij}(a\Delta\tau) \leftarrow \sum_{x=1}^{a} \frac{e^{(\eta_{ij}+\beta_{ij}x\Delta\tau)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}x\Delta\tau)}} \left(\frac{1}{\Gamma(\lambda_{i,s})\,\lambda_{i,r}^{\lambda_{i,s}}}\,(x\Delta\tau)^{\lambda_{i,s}-1}\,e^{-\frac{x\Delta\tau}{\lambda_{i,r}}}\right)\Delta\tau$
        $g^{s}_{ij}(a\Delta s) \leftarrow \sum_{x=1}^{a} \frac{e^{(\eta_{ij}+\beta_{ij}x\Delta s)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}x\Delta s)}} \left(\frac{1}{\Gamma(\lambda_{i,s})\,\lambda_{i,r}^{\lambda_{i,s}}}\,(x\Delta s)^{\lambda_{i,s}-1}\,e^{-\frac{x\Delta s}{\lambda_{i,r}}}\right)\Delta s$
        $\bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \sum_{x=b}^{c} \frac{(g^{\tau}_{ij}(a\Delta\tau)-g^{s}_{ij}(x\Delta s))\,(V_i(a\Delta\tau|\lambda_i)-V_i(x\Delta s|\lambda_i))}{1-V_i(x\Delta s|\lambda_i)}\, v_i(x\Delta s|\lambda_i)$
    end for
    $e = \epsilon + 1$
    $z \leftarrow 1$
    $\tilde{p}^{(o)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$, $\forall a, b, c, i, j$.
    while $e > \epsilon$ do
        $CQ_{i,j,k}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow$
                $\text{IFFT}\left(\text{FFT}\left(\text{diff}\left(\bar{Q}_{ik}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})\right)\right), \text{FFT}\left(\tilde{p}^{(z-1)}_{jk}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})\right)\right)$,
        $\tilde{p}^{(z)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \delta_{ij}\,\bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) + \sum_{k=1}^{N} CQ_{i,j,k}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$
        $\tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) = \left[\tilde{p}^{(z)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}))\right]_{i,j,a,b,c}$
        $e \leftarrow \left\| \tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) - \tilde{\mathbf{P}}^{(z-1)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \right\|_{\infty}$
        $z \leftarrow z + 1$
    end while
    return $\tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$
end procedure
```

It is important to note that we do not need to solve for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ during real-time inference. Instead, we create a look-up table comprising a discretized version of $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = [\tilde{p}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}))]_{i,j,a,b,c}$, and then we query this table when performing real-time inference for monitored patients. Hence, efficient and fast inferences can be provided for critical care patients for whom prompt diagnoses are necessary for the efficacy of clinical interventions. Algorithm 1 shows a pseudocode for constructing a look-up table of interval transition probabilities, `TransitionLookUp`(Γ, ϵ), which takes as an input the parameter set Γ and a precision level ϵ (to control the termination of the successive approximation iterations), and outputs the interval transitions look-up table $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$. In Algorithm 1, FFT and IFFT refer to the fast Fourier transform operation and its inverse, respectively, and "diff(.)" refers to a numerical differentiation operation.

Now that we have constructed the algorithm `TransitionLookUp` to compute the interval transition probabilities in the look-up table $\tilde{\mathbf{P}}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$, we can implement a forward-filtering inference algorithm using dynamic programming (by virtue of the recursive formula in (10)). In particular, the posterior probability of the patient's current clinical state in

Algorithm 2 Forward filtering inference

```
procedure ForwardFilter($\Gamma$, $\{y(t_w)\}_{w=1}^{m}$, $\epsilon$)
    Input: Observed samples $\{y(t_w)\}_{w=1}^{m}$, HASMM parameters $\Gamma$, and precision $\epsilon$
    Output: The posterior state distribution $\{\mathbb{P}(X(t_m) = j \,|\, \{y(t_w)\}_{w=1}^{m})\}_{j=1}^{N}$
    $\tilde{\mathbf{P}}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow$ TransitionLookUp($\Gamma$, $\epsilon$)
    $\alpha_1(j,1) = d\mathbb{P}(y(t_1) \,|\, X(t_1) = j)\, \sum_{i=1}^{N} \tilde{p}_{ij}(t_1, 0, 0) \cdot p_i^o,\ \forall j \in \mathcal{X}$
    for $j$ = 1 to $N$
    for $z = 2$ to $m$ do
        for $w = 1$ to $z$ do
            $a^*(z,w) = \arg\min_a |t_z - t_{z-w} - a\Delta\tau|$
            $b^*(z,w) = \arg\min_b |t_z - t_{z-w+1} - b\Delta\underline{s}|$
            $c^*(z,w,w^{'}) = \arg\min_c \left|t_z - t_{z-w-w'} - c\Delta\bar{s}\right|$
            $\alpha_z(j,w) = d\mathbb{P}(\{y(t_u)\}_{u=z-w+1}^{z} \,|\, X(t_z) = j)\, \sum_{i=1}^{N} \sum_{w'=1}^{z-w} \alpha_{z-w}(i,w^{'}) \times$
    $\tilde{p}_{ij}(a^*(z,w)\Delta\tau, b^*(z,w)\Delta\underline{s}, c^*(z,w,w^{'})\Delta\bar{s})\ \times (V_j(t_z - t_{z-w}|\lambda_j) - V_j(t_z - t_{z-w+1}|\lambda_j))$
        end for
    end for
    $\mathbb{P}(X(t_m) = j \,|\, \{y(t_u)\}_{u=1}^{m}) = \frac{\sum_{w=1}^{m} \alpha_m(j,w)}{\sum_{k=1}^{N}\sum_{w=1}^{m} \alpha_m(k,w)}$
    return $\{\mathbb{P}(X(t_m) = j \,|\, \{y(t_w)\}_{w=1}^{m})\}_{j=1}^{N}$
end procedure
```

terms of the forward messages can be written as

$$\mathbb{P}(X(t_m) = j \,|\, y(t_1), \ldots, y(t_m)) = \frac{\sum_{w=1}^{m} \alpha_m(j,w)}{\sum_{k=1}^{N} \sum_{w=1}^{m} \alpha_m(k,w)}. \tag{16}$$

Algorithm 2, `ForwardFilter`, implements real-time inference of a patient's clinical state given a sequence of measurements $\{y(t_1), \ldots, y(t_m)\}$. In Algorithm 2, we invoke `TransitionLookUp` initially to construct the look-up table of transition probabilities, but in practice, the look-up table can be constructed in an offline stage once the HASMM parameter set Γ is known.

The number of computations can be reduced by limiting the lags w for every forward message $\alpha_m(j,w)$ to a maximum of W lags. Ignoring the computations involved in evaluating the GP likelihoods, the complexity of `ForwardFilter` is $\mathcal{O}(mWN + mN^2)$. Since evaluating the GP likelihoods is cubic in the number of observations, the worst caset complexity of `ForwardFilter` is $\mathcal{O}((mWN + mN^2)W^3)$. (In most practical clinical settings of interest, the number of observations W can be restricted to include the most recent few samples.)

3.3 Prognostic risk scoring using an HASMM

Diagnostic inference, e.g. estimating the patient's current state after a screening test, can be conducted by a direct application of the forward filtering algorithm presented in the previous Subsection. Prognostic risk scoring plays an important role in designing screening guidelines

(Gail and Mai (2010)), acute care interventions (Knaus et al. (1985)) and surgical decisions (Foucher et al. (2007)). A risk score is a measure for the patient's risk of encountering an adverse event (abstracted as state N in our model) at any future time step starting from time t_m. That is, the patient's risk score at time t_m can be formulated as

$$\begin{aligned} R(t_m) &= \mathbb{P}(\mathcal{A}_N \,|\, \{y(t_u)\}_{u=1}^{m}, \Gamma) \\ &= 1 - \mathbb{P}(X(\infty) = N \,|\, \{y(t_u)\}_{u=1}^{m}, \Gamma), \end{aligned} \tag{17}$$

which can be computed using the outputs of `TransitionLookUp` and `ForwardFilter` as follows

$$R(t_m) = \sum_{j=1}^{N} \tilde{p}_{jN}(A,0,0) \cdot \frac{\sum_{w=1}^{m} \alpha_m(j,w)}{\sum_{k=1}^{N} \sum_{w=1}^{m} \alpha_m(k,w)}. \tag{18}$$

Therefore, the procedures `TransitionLookUp` and `ForwardFilter` suffice for executing both the diagnostic and prognostic inference tasks.

4. Learning Hidden Absorbing Semi-Markov Models

In Section 3.1, we developed an inference algorithm that can handle diagnostic and prognostic tasks for patients in real-time assuming that the true HASMM parameter set Γ is known. In practice, the parameter set Γ is not known, and has to be learned from an offline EHR dataset $\mathcal{D}$ that comprises D episodes for previously hospitalized or monitored patients, i.e.

$$\mathcal{D} = \left\{ \{y_m^d, t_m^d\}_{m=1}^{M^d}, T_c^d, l^d \right\}_{d=1}^{D},$$

where $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ are the observable variables and sampling times for the d^{th} episode, T_c^d is the episode's censoring time, and $l^d \in \{1, N\}$ is a label for the realized absorbing state.

We note that unlike the conventional HMM learning setting (Rabiner (1989); Zhang et al. (2001); Nodelman et al. (2012)), the episodes are not of equal-length as the observations for every episode stop at a random, but informative, censoring time. Thus, the patient's state trajectory does not manifest only in the observable time series, i.e. $\{y_m^d, t_m^d\}_{m=1}^{M^d}$, but also in the episode's censoring variables $\{T_c^d, l^d\}$. In this Section, we develop an efficient algorithm, which we call the *forward-filtering backward-sampling Monte Carlo EM* (FFBS-MCEM) algorithm, that computes the Maximum Likelihood (ML) estimate of Γ given an informatively censored dataset $\mathcal{D}$, i.e. $\Gamma^* = \arg\max_{\Gamma} \Lambda(\mathcal{D} \,|\, \Gamma)$, where $\Lambda(\mathcal{D} \,|\, \Gamma) = d\mathbb{P}(\mathcal{D} \,|\, \Gamma)$ is the likelihood of the dataset $\mathcal{D}$ given the parameter set Γ. We start by presenting the learning setup in Section 4.1, and then we present the FFBS-MCEM algorithm Section 4.3.

4.1 The Learning Setup

We focus on the challenging scenario when no domain knowledge or diagnostic assessments for the patients' latent states are provided in the dataset [17] $\mathcal{D}$ (with the exception of

[17] For some settings, such as chronic kidney disease progression estimation (Eddy and Neilson (2006)), the EHR records may include some anchors or assessments to the latent states over time. A simpler version

the absorbing state which is declared by the variable l^d), i.e. the learning setup is an *unsupervised* one. For such a scenario, the main challenge in constructing the ML estimator Γ^* resides in the hiddenness of the patients' state trajectories in the training dataset $\mathcal{D}$; the dataset $\mathcal{D}$ contains only the sequence of observable variables, their respective observation times, the episode's censoring time and the state in which the trajectory was absorbed. If the patients' latent state trajectories $(X(t))_{t\in\mathbb{R}_+}$ were observed in $\mathcal{D}$, the ML estimation problem $\Gamma^* = \arg\max_{\Gamma} \mathbb{P}(\mathcal{D}\,|\,\Gamma)$ would have been straightforward; the hiddenness of $(X(t))_{t\in\mathbb{R}_+}$ entails the need for marginalizing over the space of all possible latent trajectories conditioned on the observed variables, which is a hard task even for conventional CT-HMM models (Liu et al. (2015); Nodelman et al. (2012); Leiva-Murillo et al. (2011); Metzner et al. (2007)).

We start by writing the complete likelihood, i.e. the likelihood of an HASMM with a parameter set Γ to generate both the hidden states trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ and the observable variables $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ for the d^{th} episode in the dataset $\mathcal{D}$ as follows

$$d\mathbb{P}\left(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,\middle|\, \Gamma\right) = \mathbb{P}(x_1^d|\Gamma)\,\cdot\, d\mathbb{P}(s_1^d|x_1^d,\Gamma)\,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d\in\mathcal{T}_1^d}|x_1^d,\Gamma)\times$$

$$\prod_{n=2}^{k^d}\mathbb{P}(x_n^d\,|\,x_{n-1}^d, s_{n-1}^d,\Gamma)\,\cdot\, d\mathbb{P}(s_n^d\,|\,x_n^d,\Gamma)\,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d\in\mathcal{T}_n^d}\,|\,x_n^d,\Gamma), \tag{19}$$

where k^d is the number of states that realized in episode d from $t = 0$ until absorption. The factorization in (19) follows from the conditional independence properties of the HASMM variables (see Figure 6). Since we cannot observe the latent states trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$, the ML estimator deals with the expected likelihood $\Lambda(\mathcal{D}\,|\,\Gamma)$, which is evaluated by marginalizing the complete likelihood over the latent states trajectories, i.e.

$$\begin{aligned}\Lambda(\mathcal{D}\,|\,\Gamma) &= \mathbb{E}_{x(t)|\mathcal{D},\Gamma}\left[\prod_{d=1}^{D} d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d}\,|\,\Gamma)\right]\\ &= \prod_{d=1}^{D}\mathbb{E}_{x^d(t)|\mathcal{D},\Gamma}\left[\, d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d}\,|\,\Gamma)\,\right]\\ &= \prod_{d=1}^{D}\int d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d}\,|\,\Gamma)\,\cdot\, d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}\,|\,\mathcal{D},\,\Gamma), \end{aligned}\tag{20}$$

where the expectation is taken with respect to the latent trajectory conditioned on the observed dataset $\mathcal{D}$, **which contains the information on every episode's censoring time T_c^d and terminating state l^d**. The integral in (20) can be further decomposed as follows

$$\Lambda(\mathcal{D}\,|\,\Gamma) = \prod_{d=1}^{D}\int \mathbb{P}(x_1^d|\Gamma)\,\cdot\, d\mathbb{P}(s_1^d|x_1^d,\Gamma)\,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d\in\mathcal{T}_1^d}|x_1^d,\Gamma)\,\cdot\, d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}\,|\,\mathcal{D},\,\Gamma)$$

$$\times\prod_{n=2}^{k^d}\mathbb{P}(x_n^d\,|\,x_{n-1}^d, s_{n-1}^d,\Gamma)\,\cdot\, d\mathbb{P}(s_n^d\,|\,x_n^d,\Gamma)\,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d\in\mathcal{T}_n^d}\,|\,x_n^d,\Gamma). \tag{21}$$

of the learning algorithm proposed in this Section can be used to deal with such datasets. In critical care settings, it is more common that the EHR records are not labeled with any clinical state assessments over time (Yoon et al. (2016)).

4.2 Challenges Facing the HASMM Learning Task

The problem of learning the HASMM parameters by maximizing the likelihood function in (21) is obstructed by various obstacles that hinder the deployment of off-the-shelf learning algorithms; we list these challenges hereunder.

1. Finding the ML estimate Γ^* by direct maximization of $\Lambda(\mathcal{D} \,|\, \Gamma)$ is not viable due to the intractability of the integral in (21), i.e. $\Lambda(\mathcal{D} \,|\, \Gamma)$ has no analytic maximizer. The difficulty of evaluating the expected likelihood $\Lambda(\mathcal{D} \,|\, \Gamma)$ follows from the need to average the complete likelihood over a complicated posterior density function for the latent state trajectory.

2. Direct adoption of the conventional Baum-Welch implementation of the EM algorithm as a solution to the intractable problem of maximizing the expected likelihood in (21) –as has been applied in HMMs (Rabiner (1989)), HSMMs (Murphy (2002)), EDHMMs and VDHMMs (Yu (2010)– is not possible for the HASMM setting. This is due to the intractability of the integral involved in the E-step; a problem that is also faced by other continuous-time models (Liu et al. (2015); Nodelman et al. (2012)). However, these models assumed Markovian state trajectories, in which case the implementation of the E-step boils down to computing the expected state durations and transition counts as sufficient statistics for estimating the latent trajectories[18] (e.g. see Equations (12) and (13) in (Liu et al. (2015))). This simplification, which follows from the plausible properties of the Markov chain's transition rate matrix, does not materialize for semi-Markovian transitions. Further complications are introduced by the duration-dependence of the state-transitions and the segmental nature of the observables.

3. Learning an informatively censored dataset would naturally benefit from the information conveyed in the censoring variables $\{T_c^d, l^d\}$. However, the availability of censoring information leads to more complicated posterior density expressions for the latent state trajectories, which complicates the job of any analytic, variational or Monte Carlo based inference method one would use to infer the latent state trajectories[19].

In the following Section, we present a learning algorithm that addresses the above challenges, and provides insights into general settings in which informatively censored time series data are to be dealt with.

[18] Different approaches have been developed in the literature for computing these quantities: (Wang et al. (2014)) assumes that the transition rate matrix is diagonalizable, and hence utilize a closed-form estimator for the transition rates, whereas (Liu et al. (2015)) uses the *Expm* and *Unif* methods (originally developed in (Hobolth and Jensen (2011))) to evaluate the integrals of the transition matrix exponential. Unfortunately, none of these methods could be utilized for computing the proximal log-likelihood of an HASMM due to the semi-Markovianity of the state trajectory (i.e. state-durations are not exponentially distributed as it is the case in (Liu et al. (2015); Nodelman et al. (2012); Hobolth and Jensen (2011); Wang et al. (2014))).

[19] Note that the censoring information are only available in the model training (learning) phase since we deal with an offline batch of data through which we can see the full patients' episodes, whereas real-time inference, discussed in the previous Section, does not take advantage of any external censoring information.

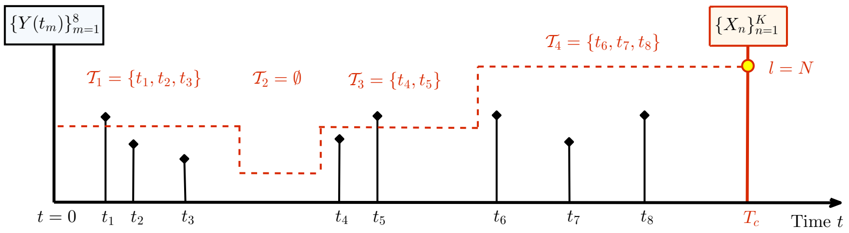

$\{Y(t_m)\}_{m=1}^{8}$
$\{X_n\}_{n=1}^{K}$
$\mathcal{T}_1 = \{t_1, t_2, t_3\}$
$\mathcal{T}_2 = \emptyset$
$\mathcal{T}_3 = \{t_4, t_5\}$
$\mathcal{T}_4 = \{t_6, t_7, t_8\}$
$l = N$
$t = 0$
t_1
t_2
t_3
t_4
t_5
t_6
t_7
t_8
T_c
Time t

Figure 9: An episode that comprised 8 observable samples, censored at time T_c, and absorbed in state N (catastrophic state). The dashed state trajectory is a trajectory that could have generated the observables with a positive probability. Computing the proximal log-likelihood requires averaging over infinitely many paths that could have generated the observables with a positive probability.

4.3 The Forward-filtering Backward-sampling Monte Carlo EM Algorithm

4.3.1 Expectation-Maximization

As in the case of classical discrete and continuous-time HMMs, we address the first challenge stated in Section 4.2 by using the EM algorithm (Liu et al. (2015); Nodelman et al. (2012)). The iterative EM algorithm starts with an initial guess $\hat{\Gamma}^o$ for the parameter set, and maximizes a proxy for the log-likelihood in the z^{th} iteration as follows:

- **E-step:** $U(\Gamma; \hat{\Gamma}^{z-1}) = \sum_{d=1}^{D} \mathbb{E}\left[\log(\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) \middle|\, \mathcal{D}, \hat{\Gamma}^{z-1}\right].$

- **M-step:** $\hat{\Gamma}^z = \arg\max_{\Gamma} U(\Gamma; \hat{\Gamma}^{z-1})$.

The E-step computes the *proximal expected log-likelihood* $U(\Gamma; \hat{\Gamma}^{z-1})$, which entails evaluating the following integral

$$U(\Gamma; \hat{\Gamma}^{z-1}) = \sum_{d=1}^{D} \int \log(d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) \cdot d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \mathcal{D}, \hat{\Gamma}^{z-1}),$$

where $d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \mathcal{D}, \hat{\Gamma}^{z-1}) = d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d, \hat{\Gamma}^{z-1})$. That is, the proximal expected log-likelihood $U(\Gamma; \hat{\Gamma}^{z-1})$ is computed by marginalizing the likelihood of the observed samples of the d^{th} episodes $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ over all potential latent paths $(x^d(t))_{t \in \mathbb{R}_+}$ that are censored at time T_c^d and absorbed in state l^d. Figure 9 depicts a set of observables $(\{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d)$ for one episode, and a potential latent path $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ that could have generated such observables. Computing $U(\Gamma; \hat{\Gamma}^{z-1})$ requires averaging over the posterior density of the latent paths conditional on an observable episode.

4.3.2 "The Only Good Monte Carlo is a Dead Monte Carlo"

Since computing $U(\Gamma; \hat{\Gamma}^{z-1})$ does not admit a closed-form solution, as mentioned earlier in the second challenge stated in Section 4.2, we resort to a Monte Carlo approach for approximating the integral involved in the E-step (Caffo et al. (2005)). That is, in the z^{th} iteration of the EM algorithm, we draw G random trajectories $(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}})_{g=1}^{G}$ for every episode

d, and use those trajectories to construct a Monte Carlo approximation $\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1})$ for the proximal log-likelihood function $U(\Gamma; \hat{\Gamma}^{z-1})$. Sample trajectories are drawn from the posterior density of the latent states' trajectory conditional on the the observable variables (including the censoring information). That is to say, the g^{th} sample trajectory $\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}$ is drawn as follows

$$\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \sim d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d, \hat{\Gamma}^{z-1}), \tag{22}$$

for $g \in \{1, \ldots, G\}$. Hence, the proximal log-likelihood $U(\Gamma; \hat{\Gamma}^{z-1})$ can be approximated via a Monte Carlo estimate $\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1})$ as follows

$$\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) \triangleq \sum_{d=1}^{D} \frac{1}{G} \sum_{g=1}^{G} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)). \tag{23}$$

It follows from the *Glivenko-Cantelli* Theorem (Durrett (2010)) that

$$||\, U(\Gamma; \hat{\Gamma}^{z-1}) - \hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) \,||_\infty = \sup_{\Gamma} |\, U(\Gamma; \hat{\Gamma}^{z-1}) - \hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) \,| \to 0 \;\; \text{a.s.},$$

and hence the Monte Carlo implementation of the E-step becomes more accurate as the sample size G increases. Sampling trajectories from the posterior distribution specified in (22) in order to obtain a Monte Carlo estimate for $U(\Gamma; \hat{\Gamma}^{z-1})$ is not a straight forward task; the sampler needs to jointly sample the states and their sojourn times taking into account the duration-dependent transitions among states, and that the number of variables sampled (number of states) $k^{d,g}$ in each trajectory is itself random.

Since there is no straightforward method that can generate samples for the random state trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ from the joint posterior density in (22), the normative solution for such a problem is to resort to a Markov Chain Monte Carlo (MCMC) method such as Metropolis-Hastings or Gibbs sampling (Carter and Kohn (1994)). Since the number of state and sojourn time variables, k^d, is itself random, one can even resort to a reversible jump MCMC method (Green and Hastie (2009)) in order to generate the samples for $\{x_n^d, s_n^d\}_{n=1}^{k^d}$. At this point of our analysis, we invoke the classical aphorism with which we titled this Subsection: *"The Only Good Monte Carlo is a Dead Monte Carlo"* (Trotter and Tukey (1956)). By this quote, Trotter meant to advocate the view that sophisticated Monte Carlo methods should be avoided whenever possible; whenever an integral is analytically tractable, or whenever some analytic insights can be exploited to built simpler samplers, doing so should be preferred to an expensive Monte Carlo method. MCMCs are indeed expensive: they mix very slowly and they generate correlated samples. Adopting an MCMC to generate random state trajectories in every iteration of the EM algorithm and for every episode in $\mathcal{D}$ is beyond affordable. Fortunately, in the rest of this Section we show that an efficient sampler that generates independent samples of $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ and for which the run-time is geometrically distributed can be constructed by capitalizing on the censoring information and utilizing some insights from the literature on sequential Monte Carlo smoothing (Godsill et al. (2004)).

4.3.3 The Forward-filtering Backward-sampling Recipe

The availability of the censoring information for every episode d in $\mathcal{D}$, together with the inherent non-linearity of the semi-Markovian transition dynamics encourage the development of a *forward-filtering backward-sampling* (FFBS) Monte Carlo algorithm[20] that goes in the reverse-time direction of every episode by starting from the censoring instance, and sequentially sampling the latent states conditioned on the (sampled) future trajectory (Godsill et al. (2004)). That is, unlike the *generative process* (described by the routine `GenerateHASMM`(Γ)) which uses the knowledge of Γ to generate sample trajectories by drawing an initial state and then sequentially going forward in time and sampling future states until absorption, the *inferential process* naturally goes the other way around: it exploits informative censoring by starting from the (known) final absorbing state (and censoring time), and sequentially samples a trajectory by traversing backwards in time and conditioning on the future.

We start constructing our forward-filter backward-sampler by first formulating the posterior density of the latent trajectory $\{x_n, s_n\}_{n=1}^{k}$ (from which we sample the G trajectories in the z^{th} iteration of the FFBS-MCEM algorithm as shown in (22)) as follows

$$d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l,\, \hat{\Gamma}^{z-1}) =$$
$$d\mathbb{P}(s_k \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) \cdot \prod_{n=1}^{k-1} d\mathbb{P}(x_n, s_n \,|\, \underbrace{\{x_{n'}, s_{n'}\}_{n'=n+1}^{k}}_{\text{Future trajectory}},\, \{y_m, t_m\}_{m=1}^{M},\, T_c), \quad (24)$$

where the conditioning on the $(z-1)^{th}$ guess of the parameter set, $\hat{\Gamma}^{z-1}$ and the episode index d are suppressed for notational convenience. The formulation in (24) decomposes the posterior density of the latent trajectory $\{x_n, s_n\}_{n=1}^{k}$ into factors in which the likelihood of every state n is conditioned on the future trajectory starting from n (i.e. the states x_{n+1} up to the absorbing states, together with their corresponding sojourn times). The posterior density in (24) can be further decomposed as follows

$$d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) = d\mathbb{P}(s_k \,|\, x_k = l, s_k < T_c) \times$$
$$\prod_{n=1}^{k-1} d\mathbb{P}(x_n, s_n \,|\, \{x_{n'}, s_{n'}\}_{n'=n+1}^{k}, s_n < \underbrace{T_c - (s_{n+1} + \ldots + s_k)}_{\text{Elapsed time in the episode}}, \{y_m, t_m\}_{m=1}^{M}), \quad (25)$$

which, using the conditional independence properties of the HASMM (see Figure 6), can be simplified as follows

$$d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) =$$
$$d\mathbb{P}(s_k \,|\, x_k = l,\, s_k < T_c) \cdot \mathbb{P}(x_1 \,|\, x_2,\, s_1 = T_c - (s_2 + \ldots + s_k),\, \{(y_m, t_m) : t_m \in \mathcal{T}_1\}) \times$$
$$\prod_{n=2}^{k-1} d\mathbb{P}(x_n,\, s_n \,|\, x_{n+1},\, s_n < \underbrace{T_c - (s_{n+1} + \ldots + s_k)}_{\text{Elapsed time in the episode}}, \underbrace{\{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\}}_{\text{Observable variables up to state } n}). \quad (26)$$

[20] The methods used in this Section are also known in the literature as *sequential Monte Carlo* or *particle filtering* methods (Godsill et al. (2004)).

From (26), we can see that for the last state in every episode, i.e. state k, we already know that $x_k = l$, and hence the randomness is only in the last state's sojourn time $s_k = l$. Contrarily, for the first state, we know that conditioned on the sojourn times of the "future states" $(s_2, \ldots, s_k)$, the sojourn time of state x_1 is equal to $T_c - \sum_{n'=2}^{k} s_{n'}$ almost surely, and hence the randomness is only in the initial state realization x_1. Generally, (26) says that a sufficient statistic for the n^{th} state and sojourn time is the future trajectory (starting from state $n+1$) summarized by: the next state, i.e. x_{n+1}, the observable variables up to state n, and the time elapsed in the episode up to state n, i.e. the duration of state n cannot exceed the difference between the censoring time T_c and the sojourn time of the future trajectory that stems from state $n+1$. This is captured by the last factor in (26), which explicitly specifies the likelihood of a joint realization for a state and its sojourn time conditioned on the future trajectory. Using Bayes' rule, we can further represent the last factor in (26) in terms of familiar quantities that are directly derived from the HASMM model parameters as follows

$$\begin{aligned} & d\mathbb{P}(x_n, s_n \,|\, x_{n+1}, s_n < T_c - (s_{n+1} + \,\ldots\, + s_K), \{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\}) \\ & \quad \propto \underbrace{\mathbb{P}(x_n \,|\, \{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\})}_{\text{Forward message}} \times \underbrace{d\mathbb{P}(s_n \,|\, x_n, s_n < T_c - (s_{n+1} + \,\ldots\, + s_k))}_{\text{Truncated sojourn time distribution}} \\ & \qquad \times \underbrace{\mathbb{P}(x_{n+1} \,|\, x_n, s_n)}_{\text{Transition function}} \;. \end{aligned} \tag{27}$$

Thus, a sampler for the latent states trajectories can be constructed using the forward messages, the HASMM's transition functions $(g_{ij}(s))_{i,j}$, and the sojourn time distributions. A compact representation for the factors in (27) is given by

$$\begin{aligned} &\textbf{(Forward messages)}\ \mathbb{P}(X_n = j \,|\, \{y_{m'}, t_{m'}\}_{m'=1}^{m}, \hat{\Gamma}^{z-1}) = \alpha_m^{z-1}(j),\ \forall 1 \leq m \leq M,\ j \in \mathcal{X}. \\ &\textbf{(Transition functions)}\ \mathbb{P}(X_{n+1} = j \,|\, X_n = i, S_n = s, \hat{\Gamma}^{z-1}) = g_{ij}^{z-1}(s), i, j \in \mathcal{X}, \\ &\textbf{(Truncated sojourn times)}\ d\mathbb{P}(S_n = s \,|\, X_n = j, S_n < \bar{s}) = \frac{v_j(s|\hat{\lambda}_j^{z-1}) \,\cdot\, \mathbf{1}_{\{s<\bar{s}\}}}{V_j(\bar{s}|\hat{\lambda}_j^{z-1})}, j \in \mathcal{X}. \end{aligned}$$

Given the representations above, we can write the last factor in (26) in the z^{th} iteration of the EM algorithm as follows

$$d\mathbb{P}(x_n, s_n \,|\, x_{n+1}, s_n < \bar{s}, \{y_m, t_m\}, \hat{\Gamma}^{z-1}) \propto \alpha_m^{z-1}(x_n) \,\cdot\, g_{x_n, x_{n+1}}^{z-1}(s_n) \,\cdot\, \frac{v_{x_n}(s_n|\hat{\lambda}_{x_n}^{z-1}) \,\cdot\, \mathbf{1}_{\{s_n \leq \bar{s}\}}}{V_{x_n}(\bar{s}|\hat{\lambda}_{x_n}^{z-1})}. \tag{28}$$

From the factor decomposition in (27), we can see that informative censoring allows us to construct a sampler for the latent state trajectories that operates sequentially in the reverse time direction by sampling from the posterior probability of every state n given the future trajectory of states that starts from state $n+1$. From (28), we note that the posterior density of the latent states conditioned on the future trajectory, from which sequential sampling is conducted, can be explicitly decomposed in terms of the HASMM parameters.

A complete recipe for the forward-filtering backward-sampling procedure for sampling trajectories from the posterior density $d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l,\, \hat{\Gamma}^{z-1})$ using the decomposition in (27) and the posterior density in (28) is provided as follows:

- **Forward filtering pass:**
  For every episode in $\mathcal{D}$, compute the forward messages $\{\alpha_m^{z-1}(j)\}$ for all time instances $t_m \in \mathcal{T}$ using the current estimate for the parameter set $\hat{\Gamma}^{z-1}$, i.e. invoke the routine $\texttt{ForwardFilter}(\hat{\Gamma}^{z-1}, \{y_m, t_m\}_{m=1}^{M}, \epsilon)$.

- **Backward sampling pass:**
  For every episode in $\mathcal{D}$, carry out the following steps:

  1. Set a dummy *placeholder index* as $k^{\#} = 1$ and set $u_{k^{\#}} = l$.
  2. Sample a Bernoulli random variable $B_{k^{\#}} \sim \text{Bernoulli}(\mathbb{P}(k = k^{\#} \,|\, \{u_k, w_k\}_{k=1}^{k^{\#}-1},\, l))$.
  3. If $B_{k^{\#}} = 0$ and $k^{\#} > 1$, sample a bi-variate random variable $(u_{k^{\#}}, w_{k^{\#}})$ using the routines `TRSampler` and `BARSampler` as follows

     $$(u_{k^{\#}},\, w_{k^{\#}}) \sim \frac{1}{\mathcal{U}} \cdot \alpha_m^{z-1}(u_{k^{\#}}) \cdot g^{z-1}_{u_{k^{\#}}, u_{k^{\#}-1}}(w_{k^{\#}}) \cdot \frac{v_{u_{k^{\#}}}(w_{k^{\#}} | \hat{\lambda}^{z-1}_{u_{k^{\#}}}) \cdot \mathbf{1}_{\{w_{k^{\#}} \leq \bar{s}\}}}{V_{u_{k^{\#}}}(\bar{s} | \hat{\lambda}^{z-1}_{u_{k^{\#}}})},$$

     where $\mathcal{U} = \sum_u \int_w \alpha_m^{z-1}(u) \cdot g^{z-1}_{u, u_{k^{\#}-1}}(w) \cdot \frac{v_u(w|\hat{\lambda}_u^{z-1}) \cdot \mathbf{1}_{\{w \leq \bar{s}\}}}{V_u(\bar{s}|\hat{\lambda}_u^{z-1})}$, $\bar{s} = T_c - \sum_{n'=1}^{k^{\#}-1} w_{n'}$, and $m = \arg\max_{m'} \{\mathcal{T} : t_{m'} \leq \bar{s}\}$.

     If $B_{k^{\#}} = 0$ and $k^{\#} = 1$, then sample $w_{k^{\#}}$ as follows

     $$w_{k^{\#}} \sim \frac{v_{u_{k^{\#}}}(w_{k^{\#}} | \hat{\lambda}^{z-1}_{u_{k^{\#}}}) \cdot \mathbf{1}_{\{w_{k^{\#}} \leq T_c\}}}{V_{u_{k^{\#}}}(T_c | \hat{\lambda}^{z-1}_{u_{k^{\#}}})}.$$

  4. If $B_{k^{\#}} = 1$, then set $w_{k^{\#}} = \bar{s}$. If $k^{\#} > 1$, then sample $u_{k^{\#}}$ as follows

     $$u_{k^{\#}} \sim \frac{d\mathbb{P}(\{y_{m'}, t_{m'}\}_{m'=1}^{m} \,|\, u_{k^{\#}}) \cdot g_{u_{k^{\#}}, u_{k^{\#}-1}}(\bar{s}) \cdot v_{u_{k^{\#}}}(\bar{s}|\hat{\lambda}^{z-1}_{u_{k^{\#}}}) \cdot \hat{p}^{o,z-1}_{u_{k^{\#}}}}{\sum_u d\mathbb{P}(\{y_{m'}, t_{m'}\}_{m'=1}^{m} \,|\, u) \cdot g_{u, u_{k^{\#}-1}}(\bar{s}) \cdot v_u(\bar{s}|\hat{\lambda}_u^{z-1}) \cdot \hat{p}_u^{o,z-1}}.$$

  5. If $B_{k^{\#}} = 0$, then increment the placeholder index $k^{\#}$ and go to step 2 and repeat the consequent steps.
  6. If $B_{k^{\#}} = 1$, then set $k = k^{\#}$ and terminate the sampling process. Set the sampled trajectory by swapping the bi-variate sequence $(u_{k^{\#}}, w_{k^{\#}})$ as follows: $(x_n, s_n) = (u_{k^{\#}-n+1}, w_{k^{\#}-n+1}), \forall n \in \{1, \ldots, k^{\#}\}$.

The forward-filtering backward-sampling procedure constitutes of a forward pass in which we compute the forward messages for all the data points in $\mathcal{D}$ using the dynamic programming algorithms presented in Section 3, and a backward pass in which these forward messages are used to sample latent state trajectories. The backward sampling procedure for every episode goes as follows. We start from the censoring time at which we know what

Algorithm 3 Truncated Rejection Sampler

```
procedure TRSampler($\Gamma$, $u$, $\bar{s}$)
    Input: A parameter set $\Gamma$, a state $u$ and a truncation threshold $\bar{s}$
    Output: A random variable $s$
    $k \leftarrow 0$
    while $k = 0$ do
        $s \sim v_u(s|\lambda_u)$
        Accept $s$ and set $k \leftarrow 1$ if $s < \bar{s}$. Reject $s$ otherwise.
    end while
    return $s$
end procedure
```

Algorithm 4 Bivariate Adaptive Rejection Sampler

```
procedure BARSampler($\{\alpha(j)\}_{j=1}^{N}$, $\Gamma$, $u^{'}$, $\bar{w}$)
    Input: A set of $N$ forward messages $\{\alpha(j)\}_{j=1}^{N}$, parameter set $\Gamma$, and a state $u^{'}$
    Output: A bivariate conditional random variable $(u, w)|u^{'}$
    $k \leftarrow 0$
    while $k = 0$ do
        $u \sim \text{Multinomial}(\alpha(1), \ldots, \alpha(N))$
        $w = \texttt{TRSampler}(\Gamma, u, \bar{w})$
        $\tilde{u} \sim \text{Multinomial}(g_{u1}(w), \ldots, g_{uN}(w))$
        Accept $(u, w)$ and set $k \leftarrow 1$ if $\tilde{u} = u^{'}$. Reject $(u, w)$ otherwise.
    end while
    return $(u, w)$
end procedure
```

state has actually materialized, i.e. the absorbing state. Since we do not know the number of states in the state trajectory, we initialize a placeholder index $k^{\#} = 1$ as an index for the absorbing state, and increment it whenever a new state is sampled. We start the sampling procedure as follows. Given the the censoring variables and the observable time series, we sample the sojourn time of the last state (the absorbing state): this is sampled from a truncated sojourn time distribution, with a truncation threshold at T_c, and a point mass at T_c with an assigned measure that is equal to the posterior probability of the absorbing state being the initial state as depicted in Figure 10. This is implemented by first sampling a Bernoulli random variable $B_{k^{\#}}$ with a success probability equal to the posterior probability of the absorbing state being the initial state, and then sampling the truncated sojourn time if $B_{k^{\#}} = 0$ using the simple rejection sample executed by the routine `TRSampler` which is provided in Algorithm 3. Having sampled the last state's sojourn time, we sample the penultimate state and its sojourn time jointly using the routine `BARSampler` (Algorithm 4) as depicted in Figure 11. The routine `BARSampler` uses a sampling algorithm, that we call the *bi-variate adaptive rejection sampler*, which jointly samples the current state and its sojourn time given the next state as follows. First, a state is sampled from a Multinomial distribution with probability masses equal to the forward messages. Next, given the sam-

Algorithm 5 A sampler for latent state trajectories

```
procedure BackwardSampling($\Gamma, \{\{\alpha^o_m(j)\}_{m,j}\}, \{y_m, t_m\}_{m=1}^M, x(T_c) = l$)
    Input: Parameter $\Gamma$, forward messages, observables, and censoring information
    Output: A sampled latent state trajectory $\{x_n, s_n\}_{n=1}^k$
    $k^\# \leftarrow 1$, $u_{k^\#} \leftarrow l$, $B_{k^\#} \sim \text{Bernoulli}(\mathbb{P}(k = k^\# | \{y_m, t_m\}_{m=1}^M, x(T_c) = l))$
    if $B_{k^\#} = 0$ then
        $w_{k^\#} = \texttt{TRSampler}(\Gamma, u_{k^\#}, T_c)$
        $k^\# \leftarrow k^\# + 1$
    else
        $w_{k^\#} = T_c$, $k = 1$, $\{x_1, s_1\} \leftarrow \{u_{k^\#}, w_{k^\#}\}$
        Terminate BackwardSampling.
    end if
    while $k^\# > 0$ do
        $B_{k^\#} \sim \text{Bernoulli}(\mathbb{P}(k = k^\# | \{u_k, w_k\}_{k=1}^{k^\#-1}, \{y_m, t_m\}_{m=1}^M))$
        $\bar{s} = T_c - \sum_{n'=1}^{k^\#-1} w_{k^\#}$
        if $B_{k^\#} = 0$ then
            $(u_{k^\#}, w_{k^\#}) \leftarrow \texttt{BARSampler}(\{\alpha^o_m(j)\}_j, \Gamma, u_{k^\#-1}, \bar{s})$
            $k^\# \leftarrow k^\# + 1$
        else
            Sample the initial state $u_{k^\#}$, set $w_{k^\#} \leftarrow \bar{s}$
            $\{x_n, s_n\} = \{u_{k^\#-n+1}, w_{k^\#-n+1}\}$, $\forall n \in \{1, \ldots, k^\#\}$
            $k^\# \leftarrow -1$
        end if
    end while
    return $\{x_n, s_n\}_{n=1}^k$
end procedure
```

pled state, we sample a sojourn time from the truncated sojourn time distribution. Finally, given the sampled state and the sampled sojourn time, we sample a dummy state from a Multinomial whose masses are equal to the transition functions, and we accept the sample only if the sampled dummy state is equal to the next state. It can be easily proven that `BARSampler` generates samples that are equal in distribution to the true state trajectory.

The backward-sampling procedure operates sequentially by invoking the `BARSampler` to generate new state and sojourn times samples conditional on the previously sampled (future) states. The process terminates whenever $B_{k^\#} = 1$, i.e. a state is sampled as an "initial state". The routine `BackwardSampling` (Algorithm 5) implements the overall backward-sampling procedure for every episode in $\mathcal{D}$. **The computational complexity of the `BackwardSampling` routine is dominated by the computation of the GP likelihood (Step 4 in the sampling procedure described above), which is cubic in the number of observations ($\mathcal{O}(W^3)$). The computations in the `BackwardSampling` procedure scales only linearly with the number of states N.**

Note that, unlike the slowly mixing MCMC methods, the backward-sampling algorithm can generate the latent state trajectory in an efficient manner, i.e. the run-time of the

Algorithm 6 Forward-filtering Backward-sampling Monte Carlo EM Algorithm

```
procedure FFBS-MCEM($\mathcal{D}$, $G$, $\epsilon$)
    Input: A dataset $\mathcal{D}$, number of Monte Carlo samples $G$, and a precision level $\epsilon$
    Output: An estimate $\hat{\Gamma}$ for the HASMM parameters
    Set an initial value $\hat{\Gamma}^o$ for the HASMM parameters
    $\{\alpha_m^{d,o}\}_{m=1}^{M_d}$ = ForwardFilter($\hat{\Gamma}^o, \{y_m^d, t_m^d\}_{m=1}^{M_d}, \epsilon$), $\forall 1 \leq d \leq D$    ▷ Forward pass
    for $d = 1$ to $D$ do    ▷ Backward pass: sample $G$ latent state trajectories
        for $g = 1$ to $G$ do
            $\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}$ = BackwardSampling($\hat{\Gamma}^o, \{y_m^d, t_m^d\}_{m=1}^{M_d}, x^d(T_c^d) = l^d$)
        end for
    end for
    $z \leftarrow 1$
    $E \leftarrow \epsilon + 1$
    while $E > \epsilon$ do
        $I_{d,g}^{z-1} \leftarrow d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \,|\, \hat{\Gamma}^{z-1})/d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \,|\, \hat{\Gamma}^o)$    ▷ Importance weights
        $\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) \cdot \frac{I_{d,g}^{z-1}}{G}$    ▷ E-step
        $\hat{\Gamma}^z = \arg\max_{\Gamma} \hat{U}_G(\Gamma;\hat{\Gamma}^{z-1})$    ▷ M-step
        $z \leftarrow z + 1$
    end while
    return $\hat{\Gamma} = \hat{\Gamma}^z$
end procedure
```

backward-sampling algorithm is stochastically dominated by a geometrically-distributed random variable with a success probability that, other than in a pathological HASMM parameter settings, would not be close to zero. Moreover, since `BackwardSampling` generates independent samples, no wasteful burn-in sampling iterations are involved in the FFBS-MCEM operation. We provide a pseudocode for the overall operation of the `FFBS-MCEM` algorithm in Algorithm 6. We omit the standard EM operations for the sake of brevity. The details of the M-step is provided in Appendix D.

In Algorithm 6, we avoid the need for running the routine `BackwardSampling` in every iteration of the EM algorithm by re-using the sampled trajectories based on the initial parameter guess $\hat{\Gamma}^o$ through the usage of importance weights in the E-step. That is, in the z^{th} iteration of the EM algorithm, we implement the E-step as follows (Booth and Hobert (1999))

$$\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) \cdot \underbrace{\frac{d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \,|\, \hat{\Gamma}^{z-1})}{d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \,|\, \hat{\Gamma}^o)}}_{\text{Importance weights}}.$$

This implementation for the E-step offers a tremendous advantage in the computational cost of `FFBS-MCEM`. By using importance weights, we need to compute the forward messages and sample the latent state trajectories only once, and then reuse the sampled trajectories in all the subsequent EM iterations.

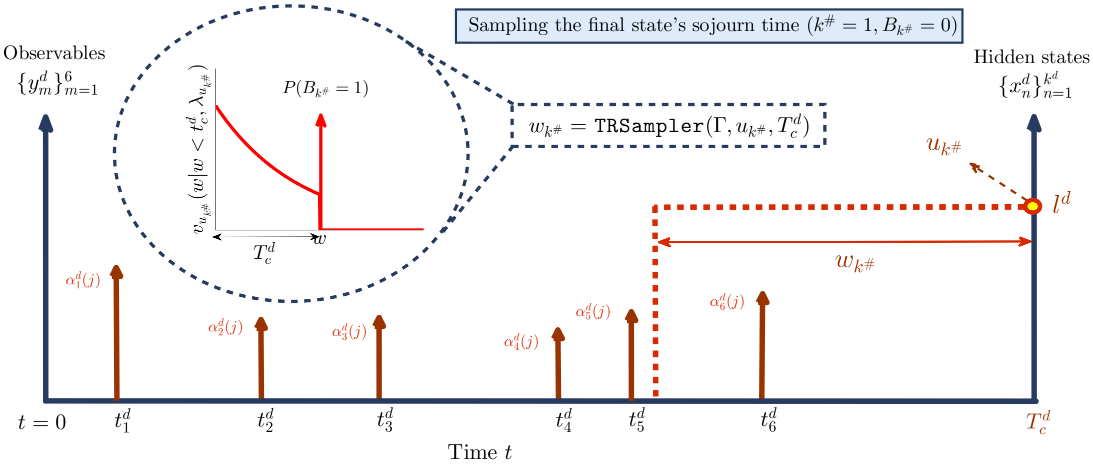


Figure 10: Depiction of the backward sampling pass for the last state of an episode d.

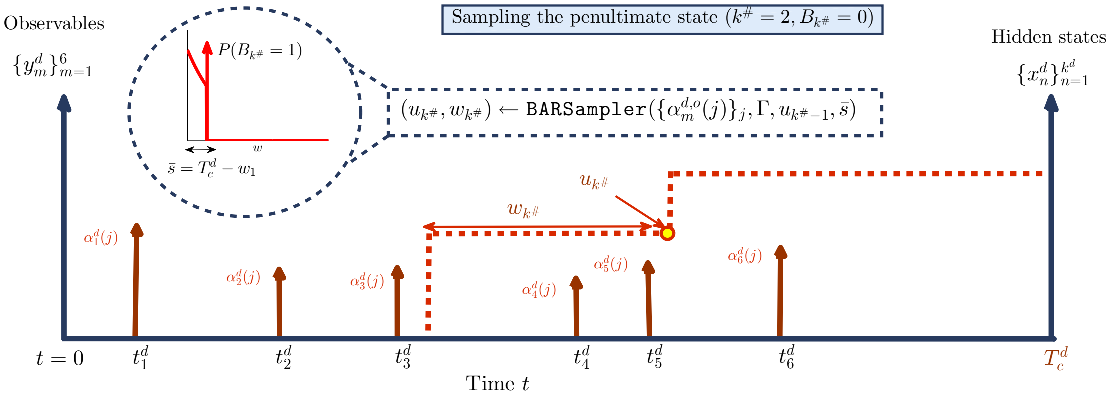


Figure 11: Depiction of the backward sampling pass for the penultimate state after having sampled the last state as depicted in the Figure above.

5. Experiments: Intensive Care Unit Prognostication

We investigate the utility of the HASMM in the setting of ICU prognostication; we use the HASMM as a model for the physiology of critically ill patients in regular hospital wards who are monitored for various vital signs and lab tests. Through the HASMM, we construct a risk score (based on the analysis in Section 3.3) that assesses the risk of clinical deterioration for the monitored patients, which allows for timely ICU admission whenever clinical decompensation is detected. Risk scoring in hospital wards and ICU admission management is a pressing problem with a huge social and clinical impact: qualitative medical studies have suggested that up to 50% of cardiac arrests on general wards could be prevented by earlier transfer to the ICU (Hershey and Fisher (1982)). Since over 200,000 in-hospital cardiac arrests occur in the U.S. each year with a mortality rate of 75% (Merchant et al. (2011)), improved patient monitoring and vigilant care in wards enabled by the HASMM would translate to a large number of lives saved yearly.

5.1 Evidence of the Clinical Utility of Early ICU Admission

Throughout this Section, we will evaluate the clinical utility of our model by investigating both the *accuracy* and *timeliness* of the real-time risk scores that the model computes. This approach has been the standard approach for evaluating the clinical utility of risk scores in retrospective clinical cohort studies that deal with critical care data (Pirracchio et al. (2015); Rothman et al. (2013). More accuracy and timeliness translates to a necessarily improved clinical outcomes; this fact has been confirmed by a large number of medical studies (Cardoso et al. (2011); Johnson et al. (2013); Hershey and Fisher (1982)). For instance, in (Cardoso et al. (2011)), it was shown that each hour of waiting in the ward was independently associated with a 1.5% increased risk of mortality in the ICU.

We stress that knowing the exact magnitude of the improvement in clinical utility (in terms of the reduction in the incidence rates of adverse outcomes) upon using our model is not possible since all available datasets are observational in nature. That is, it is impossible to answer the question of “what would have happened to the patient in the ICU had she been admitted earlier?”. Estimation of such counterfactual outcomes is also not viable due to the highly imbalanced nature of the data and the wide variety of possible adverse outcomes in the ICU, some of which are not available in our dataset. Evaluating the clinical utility in terms of the reduction in the incidence rates of adverse outcomes is only possible through an actual clinical trial. Hence, after consulting with our medical collaborators and following the clinical literature on observational studies, we rely on the accuracy and timeliness metrics as proxies for the clinical utility.

5.2 Data

5.2.1 The Patients’ Cohort

Experiments were conducted on a heterogeneous cohort of 6,094 episodes for patients who were hospitalized in Ronald Reagan UCLA medical center during the period between March 3^{rd}, 2013 to March 29^{th}, 2016. The patients’ population is heterogeneous: we considered admissions to all the floors and units in the medical center, those include the acute care pediatrics unit, cardiac observation unit, cardiothoracic unit, hematology and stem cell transplant unit and the liver transplant service. Patients admitted to those floors (or wards) are post-operative or pre-operative critically ill patients who are vulnerable to adverse clinical outcomes that may require an impending ICU transfer. The cohort comprised patients with a wide variety of ICD-9 codes and medical conditions, including leukemia, hypertension, septicemia, sepsis, abdomen and pelvis, pneumonia, and renal failure. Table 2 shows the distribution of the most common ICD-9 codes in the patient cohort together with the corresponding medical conditions . The notable heterogeneity of the cohort suggests that the results presented in this Section are generalizable to different cohorts extracted from different hospitals. Every patient in the cohort is associated with a set of 21 (temporal)

physiological streams comprising a set of vital signs and lab tests that are listed in Table 2. The physiological measurements are gathered over time during the patient’s stay in the ward, and they manifest -in a subtle fashion- the patient’s clinical state. The physiological measurements are collected over irregularly spaced time intervals (usually ranging from 1

Table 2: Characteristics of the patient cohort under study

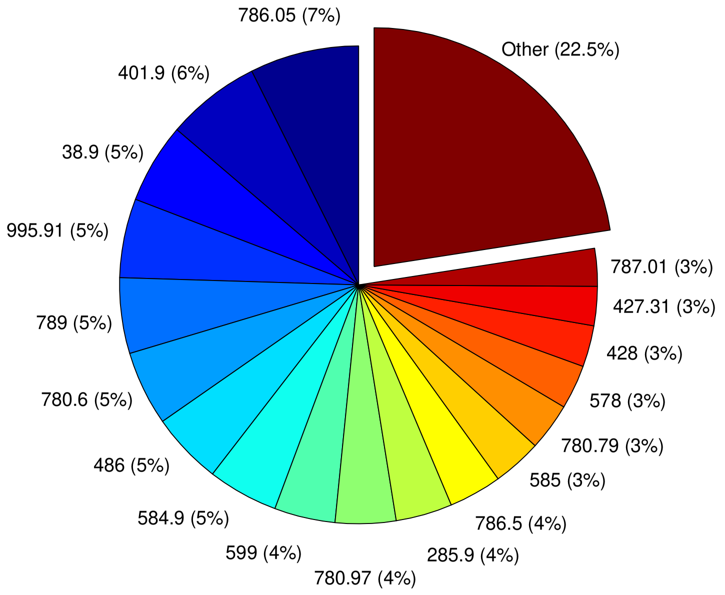
Physiological data

Vital signs	**Lab tests**
Diastolic blood pressure	Chloride
Eye opening	Glucose
Glasgow coma scale score	Urea Nitrogen
Heart rate	White blood cell count
Respiratory rate	Creatinine
Temperature	Hemoglobin
O_2 Device Assistance	Platelet Count
O_2 Saturation	Potassium
Best motor response	Saturation Sodium
Best verbal response	Total CO_2
Systolic blood pressure	

ICD-9 codes

(786.05)	Shortness of Breath
(401.9)	Hypertension
(38.9)	Septicemia
(995.91)	Sepsis
(789)	Abdomen and pelvis
(780.6)	Fever
(486)	Pneumonia
(584.9)	Renal failure
(599)	Urethra and urinary attack
(780.97)	Altered mental status
(285.9)	Anemia
(786.5)	Chest pain
(585)	Chronic renal failure
(780.79)	Malaise and fatigue
(578)	Gastrointestinal hemorrhage
(428)	Heart failure
(427.31)	Atrial fibrillation
(787.01)	Nausea

ICD-9 codes' distribution



Baseline Patient Characteristics (with 95% CI)
• **Gender distribution (Male percentage)** (Training: 50.31% ± 1.4% - Testing: 51.16% ± 2.92%)
• **Transfers from other hospitals** (Training: 11.88% ± 0.94% - Testing: 11.08% ± 1.95%)
• **Average age** (Training: 58.9 ± 0.55 years - Testing: 59.37 ± 1.11 years)
• **Patients with chemotherapy** (Training: 0.688% ± 0.272% - Testing: 1.558% ± 0.9%)
• **Patients with stem cell transplants** (Training: 0.121% ± 0.8% - Testing: 0.008% ± 0.004%)

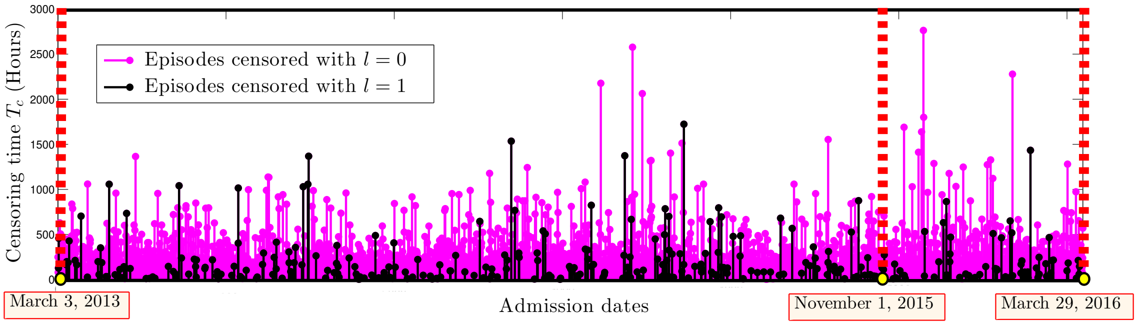


Figure 12: Visualization for the episodes' censoring information.

to 4 hours); for each physiological time series, we have access to the times at which each value was gathered.

5.2.2 Inclusion and Exclusion Criteria

In all the experiments hereafter, we split the patient cohort into a training set and a testing set. In the training set, we included a total of 4,939 patients admitted to the medical center in the period between March 3^{rd}, 2013 to November 1^{st}, 2015; the testing set comprises 1,155 patients admitted in the period between November 1^{st}, 2015 to March 29^{th}, 2016. This split of the data allows us to assess the performance under the realistic scenario when a certain algorithm learns from the data available up to a certain date, and then is used to assess the risk for patients admitted in future dates. In Table 2, we show statistics for the patients' baseline static features (e.g. gender, age, etc) in both the training and testing sets; as we can see, the characteristics of the patients admitted in the period (March 2013 - November 2015) has not significantly changed from those admitted in the period (November 2015 - March 2016). We have verified this fact using a two-sample t-test through which we compared the expected values of the baseline co-variates in both the training and testing sets. This means that the hospital's management policy with respect to the patients' acceptance and triaging has not significantly changed across the two time periods, and hence whatever is learned from the training data can be sensibly applied to the testing data. **We have excluded all patients who underwent a preplanned ICU admission from the dataset since those patients did not actually experience clinical deterioration, but were transferred routinely to the ICU after a surgery.**

5.2.3 Informative Censoring

All the patient episodes in the cohort were informatively censored. That is, for every patient in the cohort, we know the following information:

- **The censoring time (T_c):** the length of stay of each patient in the ward is recorded in the dataset, and hence we have access to the HASMM's censoring time variable T_c. The average hospitalization time (or censoring time) in the cohort is 157 hours and 34 minutes (6.5 days). The patient episodes' censoring times ranged from 4 hours to 2,672 hours.

- **The absorbing clinical state (l):** with the help of experts from the division of pulmonary and critical care medicine at Ronald Reagan UCLA medical center, we set the value of the variable l (absorbing state) for every patient's episode based on the clinicians' interventions as reported in the dataset. That is, as advised by our medical collaborators, we assigned the label $l = 1$ to every patient who was admitted to the ICU and underwent an intervention in the ICU (e.g. ventilator, drug, etc), or was reported to exhibit a cardiac or respiratory arrest (before or after the ICU transfer). According to the medical experts, those patients have experienced "clinical deterioration" as their absorbing state, and would have benefited from an earlier admission to the ICU. We assigned the label $l = 0$ to all patients who were discharged home after the clinician's in charge realized they were clinically stable. Since the readmission rate at the UCLA medical center is quite low, our medical collaborators believe that the labels $l = 1$ and $l = 0$ represent an accurate representation for the patients' true absorbing clinical states upon censoring.

Patient episodes with the absorbing state $l = 0$ had an average censoring time of 155 hours, whereas those with $l = 1$ had an average censoring time of 204 hours. The percentage of episodes with an absorbing state $l = 1$ was 4.98% ± 0.64% in the training period (March 2013 - November 2015), and was 5.19% ± 1.44% in the testing period (November 2015 - March 2016). A two-sample t-test reveals that the censoring information (distributions of T_c and l) has not significantly changed from the training to testing periods, which suggests that the HASMM learned from the training data can be sensibly applied to the testing data. Figure 12 visualizes the informative censoring information over the time period between March 2013 and March 2016. Every patient episode, starting at a certain admission date, is represented by its censoring time (hospitalization time); light colored episodes are ones that were absorbed in the clinical stability state ($l = 0$), whereas dark colored ones were absorbed in the clinical deterioration state ($l = 1$).

5.3 Baseline Algorithms

We compare our model with other baseline early warning methods. The comparisons involve both state-of-the-art clinical risk scores that are currently used in various healthcare facilities around the world, in addition to benchmark machine learning algorithms. The details of the baselines are provided in the following subsections.

5.3.1 State-of-the-art Clinical Risk Scores

We have conducted comparisons with the most prominent clinical risk scores currently deployed in major healthcare facilities. We list the clinical risk scores involved in our comparisons hereunder.

(i) **Modified Early Warning System (MEWS)**: a risk scoring scheme used currently by many healthcare facilities and rapid response teams to quickly assess the severity of illness of a hospitalized patient (Morgan et al. (1997)). The score ranges from 0 to 3 and is based on the following cardinal vital signs: systolic blood pressure, respiratory rate, SaO_2, temperature, and heart rate.

(ii) **Sequential Organ Failure Assessment (SOFA)**: a risk score (ranging from 1 to 4) that is used to determine the extent of a hospitalized patient's respiratory, cardiovascular, hepatic, coagulation, renal and neurological organ function in the ICU (Vincent et al. (1996)).

(iii) **Acute Physiology and Chronic Health Evaluation (APACHE II)**: a risk scoring system (an integer score from 0 to 71) for predicting mortality of patients in the ICU (Knaus et al. (1991)). The score is based on 12 physiological measurements, including creatinine, white blood cell count, and glasgow coma scale.

(iv) **Rothman Index**: a regression-based data-driven risk score that utilizes physiological data to predict mortality, 30-days readmission, and ICU admissions for patients in regular wards (Rothman et al. (2013)). The Rothman index is the state-of-the-art risk score for regular ward patients and is currently used in more than 70 hospitals in the US, including the Houston Methodist hospital in Texas and the Yale-New Haven hospital in Connecticut (Landro (2015)). At the time of conducting these experiments, the Rothman index was also deployed in the Ronald Reagan UCLA medical center.

We implemented the MEWS, SOFA, APACHE II and Rothman scores according to the specifications in (Vincent et al. (1996); Knaus et al. (1991); Rothman et al. (2013)). Note that while the SOFA and APACHE II scores are usually deployed for patients in the ICU, both scores have been recently shown to provide a prognostic utility for predicting clinical deterioration for patients in regular wards (Yu et al. (2014)), and hence we consider both scores in our comparisons. All the features used by these scores were also fed to the machine learning baselines.

5.3.2 Machine Learning Algorithms

In order to demonstrate the modeling gain of HASMMs, we make comparisons with 12 competing machine learning algorithms. We list all the baseline models hereunder.

- **Random forest.**
- **Logistic regression.**
- **XGBoost.**
- **AdaBoost.**
- **Bagging.**
- **Least absolute shrinkage and selection operator (LASSO).**
- **Deep Neural Networks (DNN) trained with BFGS.**
- **DNN trained with ADAM.**
- **Recurrent Neural Networks (RNN) trained with BFGS.**
- **RNN trained with ADAM.**

- **Hidden Markov Models (HMM) with Gaussian emissions.**
- **Multi-task Gaussian process (MTGP).**

Recently, notable works have built on ideas from deep learning and deep hierarchical models to construct survival predictors that learn from time-to-event data: examples of such models are those in (Katzman et al. (2016)) and (Ranganath et al. (2016)). Unfortunately, these models do not apply directly to our setting for two reasons. First, the models therein are not well-suited for handling irregularly sampled follow-ups and computing survival curves in a dynamic fashion. Second, and more importantly, our main focus is to predict whether or not a patient will exhibit clinical deterioration in the future, and not estimating survival curves with respect to a single endpoint event. Hence, our problem is technically equivalent to survival analysis with two "competing risks" (Prentice et al. (1978)), with the competing risks being *ICU admission* and *hospital discharge*, and thus the problem cannot be directly cast to the standard survival analysis setting tackled in (Katzman et al. (2016)) and (Ranganath et al. (2016)).

In order to ensure that the censoring information is properly utilized by all the discriminative predictors (Random forest, Logistic regression, XGBoost, AdaBoost, Bagging, LASSO, MTGP, and DNN), we train every predictor by constructing a training dataset that comprises the physiological data gathered within a temporal window before the censoring event (ICU admission or patient discharge), and using the censoring information (i.e. the variable l) as the labels. The size of this window is a hyper-parameter that is tuned separately for every predictor. For the testing data, the predictors are applied sequentially to a sliding window of every patient's episode, and the predictor's output is considered as the patient's real-time risk score. We used Python's `Sklearn` library (Pedregosa et al. (2011)) for training the Random forest, Logistic regression, XGBoost, AdaBoost, Bagging, LASSO and DNN predictors, and the `GPy library` (group (2012)) for training the MTGP predictor. The RNN models were implemented in `TensorFlow` (Abadi et al. (2016)).

Although RNNs are not clinically interpretable, they have been frequently applied to the problem of clinical time series prediction, and the recent work in (Che et al. (2016)) have considered RNNs to predict mortality in the ICU using the MIMIC dataset (Saeed et al. (2002)). We have trained an RNN with 5 hidden layers, and 10 neurons with each layer, using both the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm[21], where gradients are computed using the *Backpropagation Through Time* algorithm (Werbos (1990)). We have also trained an RNN model using the ADAM optimizer (Kingma and Ba (2014)). All the training time series were temporally aligned via the endpoint censoring information, and training was accomplished via 1000 iterations of the gradient descent algorithm. A top layer with a squashing sigmoid function was used to map the RNN hidden states to a risk score between 0 and 1 at each point in time. The DNNs are implemented as multi-layer perceptrons, the hyper-parameters of which (number of layers and hidden units) are optimized using grid search.

[21] We have also tried the Levenberg-Marquardt algorithm, but the network learned by BFGS offered a significantly better performance.

We used the Baum-Welch algorithm for learning the HMM (Murphy et al. (2001)); the informative censoring information was incorporated by including two absorbing states for clinical stability ($l = 0$) and deterioration ($l = 1$), and informing the forward-backward algorithm with the labeled states at the end of every episode. We tried many initializations for the HMM parameters and picked the initialization that led to the maximum likelihood for the training dataset. The complete data log likelihood after 100 EM iterations was -1.25×10^{7}. In real-time, a patient's risk score at every point of time is computed by first applying forward filtering to obtain the posterior probability of the patient's states, and then averaging over the distribution of the absorbing states. Using the Bayesian Information Criterion, we selected an HMM model with 4 latent states.

For the multi-task Gaussian process, we used the free-form parametrization (intrinsic coregionalization model) in (Bonilla et al. (2007)), and used the gradient method to learn the parameters of two Gaussian process models: one for patients with $l = 0$, and one for patients with $l = 1$. The risk score for a patient's risk score is computed as the test statistic of a sequential hypothesis test that is based on the two learned Gaussian process models. This differs from the static simulation setting in (Ghassemi et al. (2015)) were predictions are issued in a one-shot fashion using only the data obtained within 24 hours after a patient's admission.

We used the correlated feature selection algorithm to select the physiological stream for every predictor (Yu and Liu (2003)). **To ensure a fair comparison, we did not include the static (background) co-variates in any predictor, including the HASMM, since they are not used by the clinical risk scores.**

5.4 Results

5.4.1 Performance metrics

In order to assess the performance of every algorithm, we compute each algorithm's risk score $R(t)$ at every point of time in every patient's episode. We only use the patient episodes in the testing set for performance evaluation. The risk score that is based on an HASMM is evaluated as discussed in Section 3.3. We emulate the ICU admission decisions by setting a threshold on the risk score $R(t)$ above which a patient is identified as "clinically deteriorating". The accuracy of such decisions are assessed via the following performance metrics: true positive rate (TPR), positive predictive value (PPV) and timeliness. These performance metrics are formally defined as follows:

$$\text{TPR} = \frac{\#\,\text{patients with } l=1 \text{ and } R(t) \text{ exceeding threshold for some } t < Tc}{\#\,\text{patients with } l=1},$$

$$\text{PPV} = \frac{\#\,\text{patients with } l=1 \text{ and } R(t) \text{ exceeding threshold for some } t < Tc}{\#\,\text{patients with } R(t) \text{ exceeding threshold for some } t < Tc},$$

and

$$\text{Timeliness} = \mathbb{E}\left[\text{Time at which } R(t) \text{ exceeds threshold} - T_c \,|\, R(t) \text{ exceeds threshold}, l = 1\right].$$

The three performance metrics described above evaluate the different risk scoring algorithms in terms of their detection power, false alarm rate, and timeliness in detecting clinical deterioration. We sweep the threshold value of every risk scoring algorithm and report the AUC of the TPR vs. PPV ROC curve. All results reported hereafter are statistically significant (p-value < 0.001).

The usage of precision and recall (TPR and PPV) instead of the conventional (TPR and FPR) metrics is driven by the following motives. Since we are using our model as an alarm system, our algorithm only picks patients who are believed to be deterioration (patients with label 1), and we are not identifying stable patients (i.e. there is no well-defined "true negative" count). The performance of an algorithm in this particular "information retrieval" setting is more sensibly assessed via precision and recall. That is, we are trying to identify as many deteriorating patients as possible (TPR) and avoid overwhelming the ward staff with many false alarms (PPV). The "true negative" count does not play an important role in our setting. We also note that the ward's patient cohort has a significant class imbalance (the ICU admission rate is around 5%). Hence, we are typically trying to identify a small number of deteriorating patients in a large pool of stable patients. In such an unbalanced cohort, it is significantly more difficult to achieve a good PPV (PPV = TP/(TP + FP)) than a low false positive rate (FPR = FP/(FP + TN)) for a fixed TPR, since most patients are already clinically stable and therefore the false positive and true negative counts (which are counted in the stable population) will naturally be significantly larger than the true positive counts. Hence, the FPR rates (and consequently the AUC values) may look deceptively high, but they are not truly reflective of the "usefulness" of the algorithm. This is because one can still have numerous false alarms, the quantification of which is distorted by the large true negative rates that results mainly because of the fact most patients are stable. Due to reasons above, the area under the TPR vs. PPV curve has been recently identified by the critical care community as being a more sensible measure of accuracy (Romero-Brufau et al. (2015)).

While the traditional AUCROC metric can be interpreted as the probability of miss-ranking two instances with positive and negative classes, the area under the TPR vs. PPV curve can be interpreted as a measure of how well an algorithm can identify the positive classes in a pool of instances with a negative class (Davis and Goadrich (2006)). **Note that random guessing yields an area under TPR vs. FPR curve of 0.5; in the case of the TPR vs. PPV curve, and given the definitions above, random guessing yields and AUC that is equal to the fraction of instances with a positive class. That is, in our dataset, the area under the TPR vs. PPV curve for random guessing is as small as 0.05.**

5.4.2 Learning the HASMM

We applied the FFBS-MCEM algorithm to the training episodes in order to estimate the parameter set Γ. Based on the Bayesian information criterion, we have selected a model with 4 clinical states, i.e. $\mathcal{X} = \{1, 2, 3, 4\}$. State 1 is the clinical stability state, whereas state 4 is the clinical deterioration state. We ran 100 MCEM iterations and used $\hat{\Gamma}^{100}$ as the estimate for Γ. The parameter set Γ was initialized randomly using uniform distributions that cover each parameter's admissible bounds.

We discretized the time domain into steps of 1 hour while computing the elements of the look-up table holding the values of the tensor $\tilde{\mathbf{P}}$. With a granular 1-hour discretization of the time horizon, the Gaussian covariance matrix was found to be ill-conditioned for many patient episodes. To ensure the numerical stability of the computations involving the Gaussian process likelihood functions, we used the Moore-Penrose pseudo-inverse for the covariance matrix instead of direct matrix inversion. The function `TransitionLookUp` was invoked once before running the MCEM iterations, and its run time was 2 minutes and 15 seconds on a dual-core 3 GHz machine. The function `ForwardFilter` was invoked 150,852 times (all data points in all patients' episodes in both the training and testing sets), and its overall run time was 3 hours and 50 minutes (on a dual-core 3 GHz machine). The run time for every risk score update for a single patient is less than 1 second, which implies that the algorithm can efficiently prompt quick risk assessments if implemented on a machine with a reasonable computational power.

From the learned HASMM, we were able to extract the following "medical concept" out of the training data. The patients' clinical state space $\mathcal{X} = \{1, 2, 3, 4\}$ comprises the following 4 states:

- **State 1: clinical stability.**
- **State 2: type-1 critical state.**
- **State 3: type-2 critical state.**
- **State 4: clinical deterioration.**

As implied by the model, states 1 and 4 are absorbing states: once the patient is believed to be in state 1, the clinicians should release her from care, whereas exhibiting clinical state 4 should be treated with an admission to the ICU. States 2 and 3 are critical states that require the patient to stay under vigilant care in the ward. The two states are different ways to manifest "criticality". We characterize the properties of the four clinical states in the rest of this subsection.

Figures 13-16 depict the different characteristics of the four clinical states. In Figure 13, we plot a bipartite correlation graph that shows the correlations among the relevant physiological streams in the different clinical states. These graphs were constructed by computing the *Pearson correlation coefficient* $\sigma_{Y^l Y^v} = \frac{\text{cov}(Y^l, Y^v)}{\sigma_{Y^l} \cdot \sigma_{Y^v}}$ using the entries of the multi-task Gaussian process covariance matrix $\mathbf{\Sigma}$. An edge is connected between every two features for whom the Pearson correlation coefficient exceeds 0.1, i.e. $\sigma_{Y^l Y^v} > 0.1$. As we can see, different physiological variables become less or more correlated in the different clinical states. For instance, only the clinically stable patients experience significant correlations between their urea Nitrogen and the diastolic blood pressure; the Pearson coefficient between those variables becomes insignificant in the other states. Clinicians can use this piece of information, extracted solely from the data, to construct simple tests for clinical stability by computing

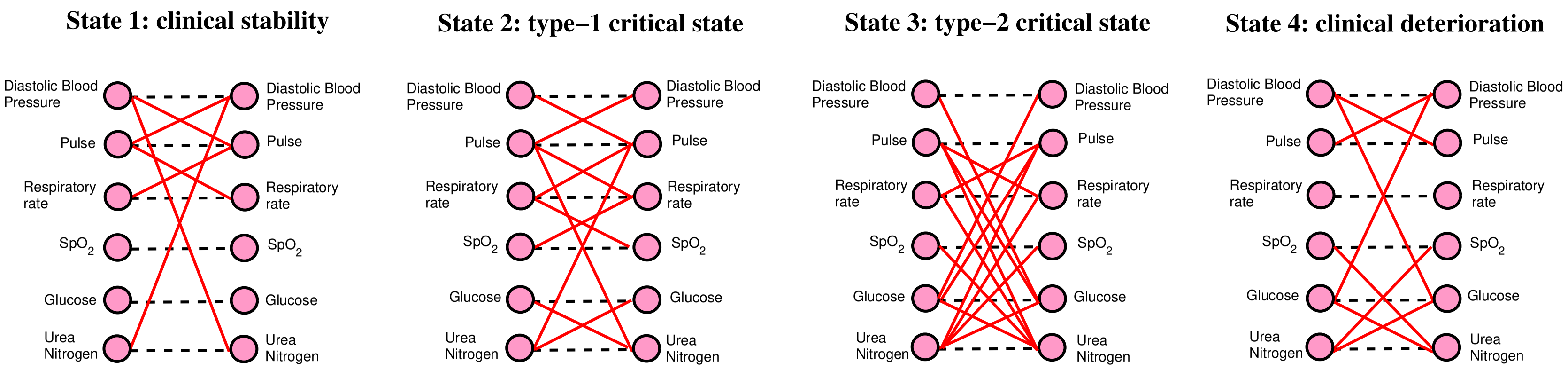

State 1: clinical stability
State 2: type–1 critical state
State 3: type–2 critical state
State 4: clinical deterioration
Diastolic Blood Pressure
Pulse
Respiratory rate
SpO$_2$
Glucose
Urea Nitrogen

Figure 13: Correlations between the patients' physiological streams in the different clinical states.

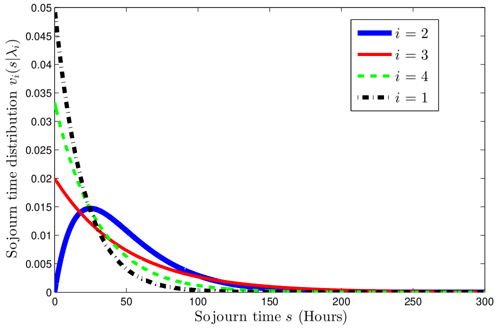


Figure 14: Sojourn time distributions.

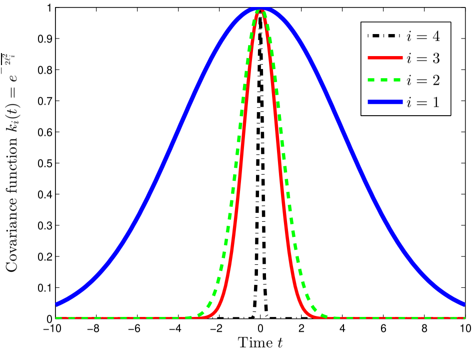


Figure 15: Covariance functions.

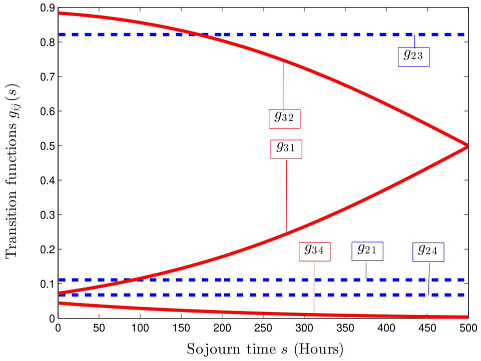


Figure 16: Transition functions.

the correlations between blood pressure and urea Nitrogen for a hospitalized patient before deciding to discharge her. Generally speaking, we observe that the critical, transient states display more correlations among the physiological streams than the clinical stability and deterioration states. In particular, the type-2 critical state has most of the physiological streams being strongly correlated. We speculate that the reason behind these strong correlations is that some kinds of interventions (e.g. drugs, mechanical pumps, ventilators, etc) applied to hospitalized patients affect all the physiological streams simultaneously; and hence we believe that type-1 and type-2 critical state patients are hospitalized patients with and without clinical interventions. We will examine this claim when we retrieve information about interventions and the time they were applied from the Ronald Reagan medical center; such information was not available at the time of conducting these experiments.

Figure 14 shows the sojourn time distributions for the four states. Recall that the "so-

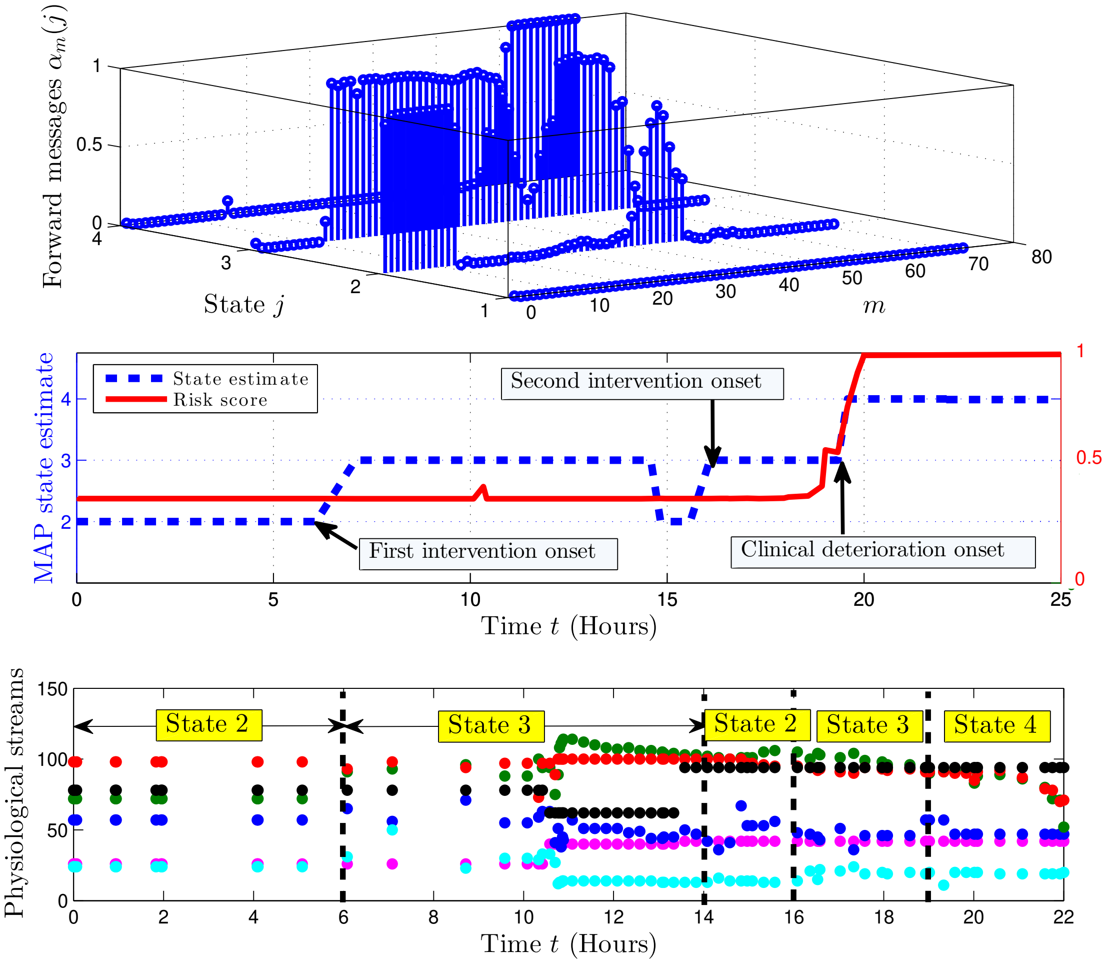


Figure 17: Depiction for the episode of a clinically deteriorating patient. (The physiological streams are color coded as follows: Diastolic blood pressure is in green, systolic blood pressure is in red, blood urea nitrogen is in cyan, and heart rate is in black, respiratory rate is in purple.)

journ time" of an absorbing state (state 1 or 4) is defined as the time between entering the state and the censoring time; such a time interval corresponds to the clinicians' policy with respect to patient discharge and ICU admission. That is, the sojourn time of an absorbing state is not a natural physiological quantity, but it rather reflects the speed with which patients are released from care or receive leveraged level of care. The sojourn time distribution for state i is an exponential distribution if the shape parameter $\lambda_{i,s} = 1$. The sojourn time distributions for states 2 and 3 significantly deviate from an exponential distribution of an ordinary, memoryless Markov model, which supports our assumption of semi-Markovianity. As we can see in Figure 14, the sojourn time distribution is not concentrated around 0 and hence is radically different from an exponential distribution (the estimated shape parameter is $\lambda_{2,s} = 2.25$). State 2 is the state with the largest first moment for the sojourn time distribution: this means that most patients in the ward exhibit this state and hence it is the most relevant for predictions. Figure 14 clearly shows that this state is not memoryless. State 3 exhibits a sojourn time distribution that is concentrated around 0; however, its shape parameter is $\lambda_{2,s} = 0.55$, and hence cannot be adequately modeled by a memoryless process.

Figure 15 displays the covariance function $k_i(t, t^{'})$ for the 4 clinical states; the state-specific covariance function quantifies the physiological streams' temporal correlations in a particular clinical conditions. Knowing such correlation patterns are useful for deciding the frequency with which nurses and clinicians should collect physiological measurements over time for different patients in different clinical conditions (Alaa and van der Schaar (2016)). We observed that, as one would expect, the temporal correlations increase when the patient becomes more stable; the temporal correlation is greatest in state 1 and smallest in state 4. This means that one would expect deteriorating patients to experience more physiological fluctuations over time. We also note that physiological stream for which the constant mean function differed significant among the clinical state was the urea Nitrogen. The level of urea Nitrogen increases significantly when the patient is in a more risky state; the average blood urea nitrogen is 11.7 milligrams per deciliter (mg/dL) in state 1, 23.8 mg/dL in state 2, 41.1 mg/dL in state 3 and 64.9 mg/dL in state 4. This is consistent with medical domain knowledge and recent discoveries in the area of critical care medicine; in (Beier et al. (2011)), it was shown that there is a substantial evidence that that elevated urea Nitrogen can be associated with all cause mortality in a heterogeneous critically ill population.

Figure 16 depicts the transition functions g_{ij} out of the transient states 2 and 3 as a function of the sojourn time in those states. We note that the transition probabilities are almost a constant function of sojourn time for patients in state 2 ($\beta_{2j} \approx 0$), whereas the duration-dependence is more significant ($\beta_{3j} > 0$); as the sojourn time in state 3 increases, the transition probabilities become more biased towards state 1. This reinforces our hypothesis that state 3 corresponds to patients for whom interventions were applied. That is, as time passes for a patient in state 3 after receiving an intervention, her chances for recovery (transiting to state 1) increases.

Now we illustrate the real-time operation of the inference algorithm as it computes risk score over time by focusing on an episode of a particular patient who was hospitalized for 1 day and then admitted to the ICU. As shown in Figure 17 (top), the inference algorithm computes the forward messages whenever new physiological measurements become available. Using the forward messages, the algorithm can display the maximum a posteriori (MAP) state estimates to the clinicians over time. As we can see in Figure 17 (middle), the patient under consideration was in clinical state 2 (type-1 critical state) at the time of admission to the ward. After 6 hours, the patient switched to state 3 (type-2 critical state), probably due to a clinical intervention. After around 9 hours, the patient switched back to the type-1 critical state for a brief 2-hour period, before switching to the type-2 critical state (probably due to a second intervention). Our algorithm was able to detect clinical deterioration (state 4) conclusively (through both the MAP state estimate and the risk score) more than 6 hours before the clinicians actually sent the patient to the ICU. Had the clinicians used the algorithm for monitoring that patient, they would have been able to send the patient to the ICU 6 hours early, allowing for a potentially much more efficient therapeutic intervention in intensive care. In Figure 17 (bottom), we plot the patient's physiological stream and tag the different time intervals with the corresponding clinical state estimates. The clinicians can rely on these clinically interpretable tags to describe the patient's states at each point of time rather than using a high-dimensional, and potentially inexpressive set of physiological measurements.

5.4.3 Performance comparisons

Since we focus on the AUC for the TPR vs. PPV performance, the AUC values are nominally less than that for the TPR vs. FPR curves. The AUC values in the TPR vs. PPV analyses are usually less than 0.5, whereas in the TPR vs. FPR analysis they can reach 0.8 (Rothman et al. (2013)). As mentioned earlier, random guessing yields an area under the TPR vs. PPV curve that is as small as 0.05. Table 3 reports the AUC and timeliness (in hours) for: ♡ HASMM, ♣ sequential (sliding-window) classification benchmarks, ♠ deep learning algorithms, ★ HMMs and ♢ clinical risk scores. As we can see, all the machine learning algorithms significantly outperform the state-of-the-art clinical risk scores (Rothman, MEWS, APACHE and SOFA). The reason behind the significant performance gain of the HASMM as compared to the clinical risk scores is that it incorporates the patients' history when updating the forward messages (as shown in Figure 17), and reasons about the future trajectory when computing the risk score (as discussed in Section 3.3). Clinical risk scores are instantaneous in that they map the current physiological measurements to a risk score without considering the previously measured physiological variables, and hence they are vulnerable to high false alarm rates (low PPV). Moreover, the clinical risk scores do not reason about the future trajectory given the current physiological measurements, and hence they display a sluggish risk signal that fail to quickly cope with subtle clinical deterioration.

With the exception of MTGPs, all the competing machine learning baselines are incapable of handling irregularly sampled data. Hence, for the baselines, we discretized the time domain into steps of 1 hour and interpolated the missing samples using zero-order-hold filtering. (We have tried cubic spline interpolation as well but this yield worse accuracies for the baselines.) The timeliness values reported in Table 3 are evaluated for the operating point for which the TPR is 50% and the PPV is 35%; this operating point was decided by

Table 3: Performance comparisons for various algorithms.

		AUC	Timeliness (hours)
♡	**HASMM**	**0.489**	**8 hrs 34 mins**
♣	Random Forest	0.362	4 hrs 21 mins
	Logistic Regression	0.271	4 hrs 36 mins
	XGBoost	0.374	7 hrs 6 mins
	AdaBoost	0.323	6 hrs 52 mins
	Bagging	0.293	6 hrs 31 mins
	MTGP	0.365	6 hrs 44 mins
	LASSO	0.261	5 hrs 21 mins
♠	RNN-BFGS	0.293	7 hrs 48 mins
	RNN-ADAM	0.311	8 hrs 6 mins
	DNN-BFGS	0.426	7 hrs 21 mins
	DNN-ADAM	0.366	6 hrs 21 mins
★	HMM	0.321	8 hrs 39 mins
♢	Rothman	0.251	—
	MEWS	0.180	—
	SOFA	0.131	—
	APACHE	0.143	—

our medical collaborators as an acceptable balance between predictive accuracy and alarm fatigue. Non of the clinical risk scores were able to achieve the desired operating point at any timeliness level.

We can see from Table 3 that HASMMs outperforms conventional HMMs; this is a consequence of incorporating temporal correlations and semi-Markovian state transitions, which more accurately describe the patient's physiology. This manifests in the sojourn time distributions in Figure 14, which largely deviate from the exponential distribution adopted by an HMM, and also manifests in the temporal correlation patterns in Figure 15, which largely deviate from the Dirac-delta function and are clearly discriminative of the different states. RNNs trained via BFGS and ADAM did not display high predictive power, probably due to the relatively limited sample size of the patient cohort under study. DNNs operating on a sliding window provided a competitive predictive accuracy: the most competitive was a DNN trained with BFGS and had its hyper-parameters (number of layers and hidden units) optimized using grid search, outperforming the different ensemble methods involved in the comparisons (XGBoost, Random Forest, AdaBoost, and Bagging).

As expected, algorithms that involved a principled time series models provided the most timely predictions as they not only evaluate the current measurements but also forecast the future. Conventional HMMs, due to their memoryless nature, provided the most timely predictions, slightly exceeding the timeliness of our model with a tight margin of 4.8 minutes. This comes at a huge false alarms' cost, manifesting in a relatively poor AUC of 0.32. This means that using a conventional HMM for risk scoring can provide decent timeliness for the patients who are identified as deteriorating, but would miss a large number of patients who will go unidentified. Contrarily, our model provides excellent timeliness together with high accuracy.

We stress that not only the proposed model outperforms the competing models in terms of accuracy, but also unlike these models, it provides a clinically interpretable model that can be used for understanding the nuances of the complex critical care setting and guiding clinical practice and ward management policies (see Figure 17). This epistemic value cannot be obtained from any of the black-box predictors, including RNNs and DNNs.

5.4.4 Controlled Analysis of the HASMM Performance

Table 3 demonstrates the performance gains achieved by our model, but it does not show the individual contributions of the different elements of the model in achieving these gains. In Table 4, we report the results of a controlled analysis of our model by investigating the impact of removing individual modeling aspects and assessing the resulting performance. To this end, we create three versions of our model, listed below, where each version lacks one of the modeling aspects:

- HASMM*: this version of the model does not capture the temporal correlations in the observations. We implement this model by forcing the length-scale parameter ℓ for all the observations to be infinite in all the iterations of the FFBS-EM algorithm. The resulting observation model corresponds to a model with independent Gaussian emissions at each time step.

- HASMM**: this version of the model adopts an exponential distribution for the sojourn times of all states. We implement this model by forcing the shape parameter $\lambda_{i,s}$ for every state i to be equal to 1.

- HASMM***: this version of the model adopts duration-independent transitions for all states. We implement this model by forcing the parameters β_{ij} to be equal to 0 for all i and j.

Table 4: A controlled analysis of the HASMM modeling aspects.

	AUC
HASMM	0.489
HASMM*	0.425
HASMM**	0.445
HASMM***	0.462

As we can see in Table 4, every aspect of the HASMM model contributes to its predictive capacity. Removing temporal correlations from our model (HASMM*) caused the biggest drop in the AUC. This shows the importance of capturing temporal correlations (i.e. physiological trends) in predicting the endpoint outcomes. The model HASMM* still significantly outperforms a standard HMM, mainly because it captures semi-Markovianity, whereas an HMM exhibits memoryless transitions, which significantly increases the false alarms. As we can see from the performance of the HASMM**, semi-Markovian transitions are instrumental in improving the AUC performance, mainly because of their role in reducing the false alarms by mitigating rapid transitions in the state process. The HASMM***, which removes duration dependence, is also inferior to the full HASMM in terms of accuracy. It is important to note that the model HASMM*** captures temporal trends in irregularly sampled data, which cannot be achieved via simpler auto-regressive HMM models that operate in discrete time.

It is important to note that our model is not developed with the exclusive goal of predicting clinical outcomes; the epistemic value of interpreting the model parameters are of great importance for managing the complex (and rather poorly understood) critical care environment. Hence, while Table 4 shows that all modeling aspects contribute to the predictive accuracy, it is worth mentioning that all of these aspects contribute to the extracted clinical knowledge as well. (This clinical knowledge is summarized in Figures 13-17.) For instance, modeling the duration dependence is not just aimed at improving the model's accuracy, but is also crucial for understating how should a clinician schedule the therapeutic interventions for a certain patient over time even during the same clinical state. A concrete example for clinical knowledge extracted from our model is the knowledge that the HASMM learned about state 3. Having learned from the model that blood urea nitrogen is predictive of clinical deterioration (see Subsection 5.4.2) and is relatively high in clinical state 3, the clinician can use the information on duration dependence in 16 to manage the administration and timing of drugs, such as Allopurinol and Aminoglycoside antibiotics, that may increase the urea nitrogen.

6. Conclusions

We developed a versatile model, which we call the Hidden Absorbing Semi-Markov Model (HASMM), for clinical time series data which accurately represents physiological data in modern EHRs. The HASMM can deal with irregularly sampled, temporally correlated, and informatively censored physiological data with non-stationary clinical state transitions. We also proposed an efficient Monte Carlo EM learning algorithms that is based on particle filtering, and developed an inference algorithm that can effectively carry out real-time inferences. We have shown, using a real-world dataset for patients admitted to the Ronald Reagan UCLA Medical Center, that HASMMs provide a significant gain in critical care prognosis when utilized for constructing an early warning and risk scoring system.

. Acknowledgments

We would like to thank Dr. Scott Hu (Division of Pulmonary and Critical Care Medicine, Department of Medicine, David Geffen School of Medicine, UCLA) for providing us with the clinical data and the appropriate medical background and insights used in Section 5. We also thank Mr. Jinsung Yoon for his valuable help with the simulations in Section 5.

This research was funded by grants from the Office of Naval Research (ONR) and NSF ECCS 1462245.

Appendix A. An Algorithm for Sampling episodes from an HASMM

Algorithm 7 Sampling episodes from an HASMM

```
procedure GenerateHASMM($\Gamma$)
    Input: HASMM model parameters $\Gamma = (N, \lambda, \mathbf{p}^o, \mathbf{Q}(s), \mathbf{\Theta}, \zeta)$
    Output: An episode $(\{X_n\}_{n=1}^{K}, \{\tau_n\}_{n=1}^{K}, \{Y(t_m)\}_{m=1}^{M}, \{t_m\}_{m=1}^{M})$
    $\tau_1 \leftarrow 0$, $k \leftarrow 1$, $\mathcal{T} \sim$ Poisson($\zeta$)                ▷ Initializations
    $x_1 \sim$ Multinomial($p_1^o, p_2^o, \ldots, p_N^o$)                ▷ Sample an initial latent state
    $s_1 \sim$ Gamma($\lambda_{x_1,s}, \lambda_{x_1,r}$), $\tau_2 \leftarrow \tau_1 + s_1$
    $\mathcal{T}_1 = \{t \in \mathcal{T} : \tau_1 \leq t \leq \tau_2\}$
    while $x_k \notin \{1, N\}$ do                ▷ Sample latent states until absorption
        $x_{k+1} \sim$ Multinomial($g_{x_k 1}(s_k), g_{x_k 2}(s_k), \ldots, g_{x_k N}(s_k)$)
        $s_{k+1} \sim$ Gamma $\left(\lambda_{x_{k+1},s}, \lambda_{x_{k+1},r}\right)$, $\tau_{k+2} \leftarrow \tau_{k+1} + s_{k+1}$
        $\mathcal{T}_{k+1} = \{t \in \mathcal{T} : \tau_{k+1} \leq t \leq \tau_{k+2}\}$
        $\{y(t_m)\}_{t_m \in \mathcal{T}_{k+1}} \sim \mathcal{GP}(\Theta_{x_{k+1}})$    ▷ Sample observations from a Gaussian Process
        $k \leftarrow k + 1$
    end while
    return $(\{x_n\}_{n=1}^{K}, \{\tau_n\}_{n=1}^{K}, \{y(t_m)\}_{m=1}^{M}, \{t_m\}_{m=1}^{M})$
end procedure
```

Appendix B. Proof of Theorem 1

We start by rewriting (11) as follows:

$$\begin{bmatrix} \tilde{p}_{11}(\tau, \underline{s}, \bar{s}) & \ldots & \tilde{p}_{1N}(\tau, \underline{s}, \bar{s}) \\ \vdots & \ddots & \vdots \\ \tilde{p}_{N1}(\tau, \underline{s}, \bar{s}) & \ldots & \tilde{p}_{NN}(\tau, \underline{s}, \bar{s}) \end{bmatrix} = \begin{bmatrix} 1 - \bar{Q}_1(\tau, \underline{s}, \bar{s}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 - \bar{Q}_N(\tau, \underline{s}, \bar{s}) \end{bmatrix} +$$
$$\int_{u=0}^{\tau} \left(\frac{\partial}{\partial u} \begin{bmatrix} \bar{Q}_{11}(u, \underline{s}, \bar{s}) & \ldots & \bar{Q}_{1N}(u, \underline{s}, \bar{s}) \\ \vdots & \ddots & \vdots \\ \bar{Q}_{N1}(u, \underline{s}, \bar{s}) & \ldots & \bar{Q}_{NN}(u, \underline{s}, \bar{s}) \end{bmatrix} \right) \times \begin{bmatrix} \tilde{p}_{11}(\tau - u, 0, 0) & \ldots & \tilde{p}_{1N}(\tau - u, 0, 0) \\ \vdots & \ddots & \vdots \\ \tilde{p}_{N1}(\tau - u, 0, 0) & \ldots & \tilde{p}_{NN}(\tau - u, 0, 0) \end{bmatrix} du. \tag{29}$$

Starting with the left hand side, we can use a first-step analysis to write every term $\tilde{p}_{ij}(\tau, \underline{s}, \bar{s})$ as follows

$$\begin{aligned}
\tilde{p}_{ij}(\tau, \underline{s}, \bar{s}) &= \mathbb{P}(X(t+\tau) = j | X(t) = i, \underline{s} \leq S(t) \leq \bar{s}) \\
&= \delta_{ij} \left(\mathbb{P}(S_i < \tau | X(t) = i, \underline{s} \leq S(t) \leq \bar{s})\right) + \\
&\int_{u=0}^{\tau} \mathbb{P}(X(t+u) = k | X(t) = i, \underline{s} \leq S(t) \leq \bar{s}) \cdot \mathbb{P}(X(t+\tau) = j | X(t+u) = k) \, du \\
&= \delta_{ij} \left(1 - \bar{Q}_i(\tau, \underline{s}, \bar{s})\right) + \\
&\int_{u=0}^{\tau} \mathbb{P}(X(t+u) = k | X(t) = i, \underline{s} \leq S(t) \leq \bar{s}) \cdot \mathbb{P}(X(t+\tau-u) = j | X(t) = k) \, du \\
&= \delta_{ij} \left(1 - \bar{Q}_i(\tau, \underline{s}, \bar{s})\right) + \int_{u=0}^{\tau} \frac{\partial}{\partial u} \sum_{k \neq i} \bar{Q}_{ik}(u, \underline{s}, \bar{s}) \cdot \tilde{p}_{kj}(\tau - u, 0, 0) \, du, \qquad (30)
\end{aligned}$$

$\forall i, j \in \mathcal{X}$, where $S(t)$ is the time elapsed in state $X(t)$, and S_i is the sojourn time of state i. The integral equation in (30) can be written in a matrix form as in the right hand side of (29), and hence the Theorem follows.

Appendix C. Proof of Theorem 2

Recall that the operation

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}))$$

can be written as

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathbf{I}_{N \times N} - \operatorname{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \left(\frac{\partial \bar{\mathbf{Q}}(., \underline{s}, \bar{s})}{\partial u} \star \tilde{\mathbf{P}}(., 0, 0)\right)(\tau).$$

Now consider n applications of the operator $\mathcal{B}(.)$, we have that

$$\begin{aligned}
\left(\frac{\partial \bar{\mathbf{Q}}(., \underline{s}, \bar{s})}{\partial u_1} \star \ldots \star \frac{\partial \bar{\mathbf{Q}}(., \underline{s}, \bar{s})}{\partial u_n} \star \tilde{\mathbf{P}}(., 0, 0)\right)(\tau) &\leq N^n \cdot \int_0^{\tau} \int_0^{\tau - u_{n-1}} \cdots \int_0^{\tau - u_1} du_1 \, du_2 .., du_n \\
&= N^n \cdot \frac{\tau^n}{n!}. \qquad (31)
\end{aligned}$$

Thus, for every $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) \in \mathcal{P}$ and every $\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s}) < 1$, there exists n such that $\mathcal{B}^n\{.\}(.)$ is a contraction mapping. Therefore, the operation $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}))$ has a unique fixed point that can be reached via $n \in \mathbb{N}$ successive approximations.

Appendix D. The M-step of the FFBS-MCEM Algorithm

The proximal likelihood function at the z^{th} iteration is given by

$$\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) \cdot \frac{I_{d,g}^{z-1}}{G},$$

where the likelihood inside the logarithm can be factorized as follows

$$d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma) = \mathbb{P}(x_1^{d,g}|\Gamma) \,\cdot\, d\mathbb{P}(s_1^{d,g}|x_1^{d,g}, \Gamma) \,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_1^d}|x_1^{d,g}, \Gamma) \times \prod_{n=2}^{k^{d,g}} \mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma) \,\cdot\, d\mathbb{P}(s_n^{d,g} \,|\, x_n^{d,g}, \Gamma) \,\cdot\, d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma),$$

and hence the log-likelihood is given by

$$\log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) = \log(\mathbb{P}(x_1^{d,g}|\Gamma)) + \sum_{n=2}^{k^{d,g}} \log(\mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma)) + \sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(s_n^{d,g} \,|\, x_n^{d,g}, \Gamma)) + \sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma)).$$

The updated parameter set $\hat{\Gamma}^z$ is obtained by maximizing:

$$\hat{\Gamma}^z = \arg\max_{\Gamma} \hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}).$$

Updating the initial state distribution is straightforwardly conducted as follows

$$\hat{p}_k^{o,z} = \frac{1}{G \cdot D} \sum_{d,g} \mathbf{1}_{\{x_1^{d,g} = k\}} \cdot \frac{I_{d,g}^{z-1}}{G}.$$

For the k^{th} state sojourn time distribution parameters (shape and rate parameters for the Gamma distribution), we solve the optimization problem by computing the Maximum Likelihood Estimate (MLE) based on the observed sojourn times in the sampled trajectories as follows

$$\begin{aligned}
\mathcal{N}_k &= \{(d,g,n) : x_n^{d,g} = k\}, \\
\Xi_k &= \log\left(\frac{1}{|\mathcal{N}_k|} \sum_{(d,g,n)\in\mathcal{N}_k} s_n^{d,g}\right) - \frac{1}{|\mathcal{N}_k|} \sum_{(d,g,n)\in\mathcal{N}_k} \log\left(s_n^{d,g}\right), \\
\hat{\lambda}_{k,s}^z &= \frac{3 - \Xi_k + \sqrt{(\Xi_k - 3)^2 - 24\Xi_k}}{12\Xi_k}, \\
\hat{\lambda}_{k,r}^z &= \frac{\hat{\lambda}_{k,s}^z \, |\mathcal{N}_k|}{\sum_{(d,g,n)\in\mathcal{N}_k} s_n^{d,g}}.
\end{aligned}$$

The optimization problem is intractable for the rest of the parameters, and hence we resort to approximate solutions. For the transition parameters, we maximize the term $\sum_{n=2}^{k^{d,g}} \log(\mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma))$ using the successive approximations, whereas for the GP parameters, we maximize the term $\sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma))$ via conjugate gradient descent.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

Ahmed M. Alaa and Mihaela van der Schaar. Balancing suspense and surprise: Timely decision making with endogenous information acquisition. In *Advances in Neural Information Processing Systems*, pages 2910–2918, 2016.

Ahmed M Alaa, Jinsung Yoon, Scott Hu, and Mihaela van der Schaar. Personalized risk scoring for critical care prognosis using mixtures of gaussian processes. *arXiv preprint arXiv:1610.08853*, 2016.

Ahmed M. Alaa, Scott Hu, and Mihaela van der Schaar. Learning from clinical judgments: Semi-markov-modulated marked hawkes processes for risk prognosis. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Jeffrey A Bakal, Finlay A McAlister, Wei Liu, and Justin A Ezekowitz. Heart failure re-admission: measuring the ever shortening gap between repeat heart failure hospitalizations. *PloS one*, 9(9):e106494, 2014.

Jirina Bartkova, Zuzana Hořejší, Karen Koed, Alwin Krämer, Frederic Tort, Karsten Zieger, Per Guldberg, Maxwell Sehested, Jahn M Nesland, Claudia Lukas, et al. Dna damage response as a candidate anti-cancer barrier in early human tumorigenesis. *Nature*, 434 (7035):864–870, 2005.

Kevin Beier, Sabitha Eppanapally, Heidi S Bazick, Domingo Chang, Karthik Mahadevappa, Fiona K Gibbons, and Kenneth B Christopher. Elevation of bun is predictive of long-term mortality in critically ill patients independent of'normal'creatinine. *Critical care medicine*, 39(2):305, 2011.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2007.

James G Booth and James P Hobert. Maximizing generalized linear mixed model likelihoods with an automated monte carlo em algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):265–285, 1999.

Brian S Caffo, Wolfgang Jank, and Galin L Jones. Ascent-based monte carlo expectation–maximization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):235–251, 2005.

Lucienne TQ Cardoso, Cintia MC Grion, Tiemi Matsuo, Elza HT Anami, Ivanil AM Kauss, Ludmila Seko, and Ana M Bonametti. Impact of delayed admission to intensive care units on mortality of critically ill patients: a cohort study. *Critical Care*, 15(1):R28, 2011.

Chris K Carter and Robert Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

Dustin Charles, Meghan Gabriel, and JaWanna Henry. Electronic capabilities for patient engagement among us non-federal acute care hospitals: 2012-2014. *The Office of the National Coordinator for Health Information Technology*, 2015.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

Baojiang Chen and Xiao-Hua Zhou. Non-homogeneous markov process models with informative observations with an application to alzheimer's disease. *Biometrical Journal*, 53 (3):444–463, 2011.

Jill M Cholette, Kelly F Henrichs, George M Alfieris, Karen S Powers, Richard Phipps, Sherry L Spinelli, Michael Swartz, Francisco Gensini, L Eugene Daugherty, Emily Nazarian, et al. Washing red blood cells and platelets transfused in cardiac surgery reduces post-operative inflammation and number of transfusions: Results of a prospective, randomized, controlled clinical trial. *Pediatric critical care medicine: a journal of the Society of Critical Care Medicine and the World Federation of Pediatric Intensive and Critical Care Societies*, 13(3), 2012.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Zelalem Getahun Dessie. Multi-state models of hiv/aids by homogeneous semi-markov process. *American Journal of Biostatistics*, 4(2):21, 2014.

Michael Dewar, Chris Wiggins, and Frank Wood. Inference in hidden markov models with explicit state duration distributions. *IEEE Signal Processing Letters*, 19(4):235–238, 2012.

Rick Durrett. *Probability: theory and examples.* Cambridge university press, 2010.

Allison A Eddy and Eric G Neilson. Chronic kidney disease progression. *Journal of the American Society of Nephrology*, 17(11):2964–2966, 2006.

Yohann Foucher, Eve Mathieu, Philippe Saint-Pierre, J Durand, and J Daures. A semi-markov model based on generalized weibull distribution with an illustration for hiv disease. *Biometrical journal*, 47(6):825, 2005.

Yohann Foucher, Magali Giral, Jean-Paul Soulillou, and Jean-Pierre Daures. A semi-markov model for multistate and interval-censored data with multiple terminal events. application in renal transplantation. *Statistics in medicine*, 26(30):5381–5393, 2007.

Yohann Foucher, M Giral, JP Soulillou, and JP Daures. A flexible semi-markov model for interval-censored data and goodness-of-fit testing. *Statistical methods in medical research*, 2008.

Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011a.

Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011b.

Mitchell H Gail and Phuong L Mai. Comparing breast cancer risk assessment models. *Journal of the National Cancer Institute*, 102(10):665–668, 2010.

Valentine Genon-Catalot, Thierry Jeantheau, Catherine Larédo, et al. Stochastic volatility models as hidden markov models and statistical applications. *Bernoulli*, 6(6):1051–1079, 2000.

Konstantinos Georgatzis, Christopher KI Williams, and Christopher Hawthorne. Input-output non-linear dynamical systems applied to physiological condition monitoring. *Journal of Machine Learning Research*, 2016.

Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.

Marzyeh Ghassemi, Marco AF Pimentel, Tristan Naumann, Thomas Brennan, David A Clifton, Peter Szolovits, and Mengling Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 446. NIH Public Access, 2015.

Giacomo Giampieri, Mark Davis, and Martin Crowder. Analysis of default data using hidden markov models. *Quantitative Finance*, 5(1):27–34, 2005.

Florence Gillaizeau, Etienne Dantan, Magali Giral, and Yohann Foucher. A multistate additive relative survival semi-markov model. *Statistical methods in medical research*, page 0962280215586456, 2015.

Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

Peter J Green and David I Hastie. Reversible jump mcmc. *Genetics*, 155(3):1391–1403, 2009.

Sheffield ML group. Gpy: A gaussian process framework in python. 2012.

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic markov models. In *AISTATS*, volume 7, pages 163–170, 2007.

Yann Guédon. Exploring the state sequence space for hidden markov and semi-markov chains. *Computational Statistics & Data Analysis*, 51(5):2379–2409, 2007.

Chantal Guihenneuc-Jouyaux, Sylvia Richardson, and Ira M Longini. Modeling markers of disease progression by a hidden markov process: application to characterizing cd4 cell decline. *Biometrics*, 56(3):733–741, 2000.

Tracy D Gunter and Nicolas P Terry. The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. *Journal of medical Internet research*, 7(1):e3, 2005.

Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

CharlesO Hershey and Linda Fisher. Why outcome of cardiopulmonary resuscitation in general wards is poor. *The Lancet*, 319(8262):31–34, 1982.

Asger Hobolth and Jens Ledet Jensen. Summary statistics for endpoint-conditioned continuous-time markov chains. *Journal of Applied Probability*, pages 911–924, 2011.

Helen Hogan, Frances Healey, Graham Neale, Richard Thomson, Charles Vincent, and Nick Black. Preventable deaths due to problems in care in english acute hospitals: a retrospective case record review study. *BMJ quality & safety*, pages bmjqs–2012, 2012.

William Hoiles and Mihaela van der Schaar. A non-parametric learning method for confidently estimating patient's clinical state and dynamics. In *Advances in Neural Information Processing Systems*, pages 2020–2028, 2016.

George Hripcsak, David J Albers, and Adler Perotte. Parameterizing time in electronic health record studies. *Journal of the American Medical Informatics Association*, 22(4): 794–804, 2015.

Xuelin Huang and Robert A Wolfe. A frailty model for informative censoring. *Biometrics*, 58(3):510–520, 2002.

Aparna V Huzurbazar. Multistate models, flowgraph models, and semi-markov processes. 2004.

Christopher H Jackson, Linda D Sharples, Simon G Thompson, Stephen W Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(2):193–209, 2003.

Jacques Janssen and R De Dominicis. Finite non-homogeneous semi-markov processes: Theoretical and computational aspects. *Insurance: Mathematics and Economics*, 3(3): 157–165, 1984.

Daniel W Johnson, Ulrich H Schmidt, Edward A Bittner, Benjamin Christensen, Retsef Levi, and Richard M Pino. Delay of transfer from the intensive care unit: a prospective observational study of incidence, causes, and financial impact. *Critical Care*, 17(4):R128, 2013.

Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(Feb):673–701, 2013.

Pierre Joly and Daniel Commenges. A penalized likelihood approach for a progressive three-state model with censored and truncated data: Application to aids. *Biometrics*, 55(3): 887–890, 1999.

Jared Katzman, Uri Shaham, Jonathan Bates, Alexander Cloninger, Tingting Jiang, and Yuval Kluger. Deep survival: A deep cox proportional hazards network. *arXiv preprint arXiv:1606.00931*, 2016.

Juliane Kause, Gary Smith, David Prytherch, Michael Parr, Arthas Flabouris, Ken Hillman, et al. A comparison of antecedents to cardiac arrests, deaths and emergency intensive care admissions in australia and new zealand, and the united kingdom—the academia study. *Resuscitation*, 62(3):275–282, 2004.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lisa L Kirkland, Michael Malinchoc, Megan O'Byrne, Joanne T Benson, Deanne T Kashiwagi, M Caroline Burton, Prathibha Varkey, and Timothy I Morgenthaler. A clinical deterioration prediction tool for internal medicine patients. *American Journal of Medical Quality*, 28(2):135–142, 2013.

William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. Apache ii: a severity of disease classification system. *Critical care medicine*, 13(10):818–829, 1985.

William A Knaus, Douglas P Wagner, Elizabeth A Draper, Jack E Zimmerman, Marilyn Bergner, Paulo G Bastos, Carl A Sirio, Donald J Murphy, Ted Lotring, and Anne Damiano. The apache iii prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults. *Chest Journal*, 100(6):1619–1636, 1991.

Vidyadhar G Kulkarni. *Modeling and analysis of stochastic systems.* CRC Press, 1996.

Stephan W Lagakos, Charles J Sommer, and Marvin Zelen. Semi-markov models for partially censored data. *Biometrika*, 65(2):311–317, 1978.

David Lando. On cox processes and credit risky securities. *Review of Derivatives research*, 2(2-3):99–120, 1998.

Laura Landro. Hospitals find new ways to monitor patients 24/7. *The Wall Street Journal*, 2015.

Jose Leiva-Murillo, AA Rodrguez, and E Baca-Garca. Visualization and prediction of disease interactions with continuous-time hidden markov models. In *NIPS 2011 Workshop on Personalized Medicine*, 2011.

H Lehman Li-wei, Shamim Nemati, Ryan P Adams, George Moody, Atul Malhotra, and Roger G Mark. Tracking progression of patient state of health in critical care using inferred shared dynamics in physiological time series. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7072–7075. IEEE, 2013.

William A Link. A model for informative censoring. *Journal of the American Statistical Association*, 84(407):749–752, 1989.

Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine Learning for Healthcare Conference*, pages 253–270, 2016.

Yu-Ying Liu, Shuang Li, Fuxin Li, Le Song, and James M Rehg. Efficient learning of continuous-time hidden markov models for disease progression. In *Advances in neural information processing systems*, pages 3600–3608, 2015.

Sergio Matos, Surinder S Birring, Ian D Pavord, and H Evans. Detection of cough signals in continuous audio recordings using hidden markov models. *IEEE Transactions on Biomedical Engineering*, 53(6):1078–1083, 2006.

Raina M Merchant, Lin Yang, Lance B Becker, Robert A Berg, Vinay Nadkarni, Graham Nichol, Brendan G Carr, Nandita Mitra, Steven M Bradley, Benjamin S Abella, et al. Incidence of treated cardiac arrest in hospitalized patients in the united states. *Critical care medicine*, 39(11):2401, 2011.

Philipp Metzner, Illia Horenko, and Christof Schütte. Generator estimation of markov jump processes based on incomplete observations nonequidistant in time. *Physical Review E*, 76(6):066702, 2007.

Rui P Moreno, Philipp GH Metnitz, Eduardo Almeida, Barbara Jordan, Peter Bauer, Ricardo Abizanda Campos, Gaetano Iapichino, David Edbrooke, Maurizia Capuzzo, Jean-Roger Le Gall, et al. Saps 3-from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission. *Intensive care medicine*, 31(10):1345–1355, 2005.

RJM Morgan, F Williams, and MM Wright. An early warning scoring system for detecting developing critical illness. *Clin Intensive Care*, 8(2):100, 1997.

DR Mould. Models for disease progression: new approaches and uses. *Clinical Pharmacology & Therapeutics*, 92(1):125–131, 2012.

Kevin Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001.

Kevin P Murphy. Hidden semi-markov models (hsmms). *unpublished notes*, 2, 2002.

Uri Nodelman, Christian R Shelton, and Daphne Koller. Expectation maximization and complex duration distributions for continuous time bayesian networks. *arXiv preprint arXiv:1207.1402*, 2012.

Zdzisław Opial. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bulletin of the American Mathematical Society*, 73(4):591–597, 1967.

Mari Ostendorf, Vassilios V Digalakis, and Owen A Kimball. From hmm's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on speech and audio processing*, 4(5):360–378, 1996.

Soren Erik Pedersen, Suzanne S Hurd, Robert F Lemanske, Allan Becker, Heather J Zar, Peter D Sly, Manuel Soto-Quiroz, Gary Wong, and Eric D Bateman. Global strategy for the diagnosis and management of asthma in children 5 years and younger. *Pediatric pulmonology*, 46(1):1–17, 2011.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Romain Pirracchio, Maya L Petersen, Marco Carone, Matthieu Resche Rigon, Sylvie Chevret, and Mark J van der Laan. Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

Andrei D Polyanin and Alexander V Manzhirov. *Handbook of integral equations*. CRC press, 2008.

Ross L Prentice, John D Kalbfleisch, Arthur V Peterson Jr, Nancy Flournoy, Vern T Farewell, and Norman E Breslow. The analysis of failure times in the presence of competing risks. *Biometrics*, pages 541–554, 1978.

Zhen Qin and Christian R Shelton. Auxiliary gibbs sampling for inference in piecewise-constant conditional intensity models. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 2015.

Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning for Healthcare Conference*, pages 101–114, 2016.

Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

Santiago Romero-Brufau, Jeanne M Huddleston, Gabriel J Escobar, and Mark Liebow. Why the c-statistic is not informative to evaluate early warning scores and what metrics to use. *Critical Care*, 19(1):285, 2015.

Michael J Rothman, Steven I Rothman, and Joseph Beals. Development and validation of a continuous measure of patient condition using the electronic medical record. *Journal of biomedical informatics*, 46(5):837–848, 2013.

Mohammed Saeed, Christine Lieu, Greg Raber, and Roger G Mark. Mimic ii: a massive temporal icu patient database to support research in intelligent patient monitoring. In *Computers in Cardiology, 2002*, pages 641–644. IEEE, 2002.

Daniel O Scharfstein and James M Robins. Estimation of the failure time distribution in the presence of informative censoring. *Biometrika*, 89(3):617–634, 2002.

Peter Schulam and Suchi Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*, pages 748–756, 2015.

Padhraic Smyth. Hidden markov models for fault detection in dynamic systems. *Pattern recognition*, 27(1):149–164, 1994.

Henry T Stelfox, Brenda R Hemmelgarn, Sean M Bagshaw, Song Gao, Christopher J Doig, Cheri Nijssen-Jordan, and Braden Manns. Intensive care unit bed availability and outcomes for hospitalized patients with sudden clinical deterioration. *Archives of internal medicine*, 172(6):467–474, 2012.

CP Subbe, M Kruger, P Rutherford, and L Gemmel. Validation of a modified early warning score in medical admissions. *Qjm*, 94(10):521–526, 2001.

MJ Sweeting, VT Farewell, and D De Angelis. Multi-state markov models for disease progression in the presence of informative examination times: An application to hepatitis c. *Statistics in medicine*, 29(11):1161–1174, 2010.

S Taghipour, D Banjevic, AB Miller, N Montgomery, AKS Jardine, and BJ Harvey. Parameter estimates for invasive breast cancer progression in the canadian national breast screening study. *British journal of cancer*, 108(3):542–548, 2013.

Hale F Trotter and John W Tukey. Conditional monte carlo for normal samples. In *Symposium on Monte Carlo Methods*, pages 64–79. Wiley, 1956.

John Varga, Christopher P Denton, and Fredrick M Wigley. *Scleroderma: From pathogenesis to comprehensive management*. Springer Science & Business Media, 2012.

J-L Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, CK Reinhart, PeterM Suter, and LG Thijs. The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. *Intensive care medicine*, 22(7):707–710, 1996.

Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

J Yoon, A Alaa, S Hu, and M van der Schaar. Forecasticu: A prognostic decision support system for timely prediction of intensive care unit admission. pages 1680–1689, 2016.

Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

Shun Yu, Sharon Leung, Moonseong Heo, Graciela J Soto, Ronak T Shah, Sampath Gunda, and Michelle Ng Gong. Comparison of risk prediction scoring systems for ward patients: a retrospective nested case-control study. *Critical Care*, 18(3):1, 2014.

Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.