Latent State Space Models for Prediction

by

Prashan Wanigasekara

Submitted to the Systems Design and Management Program
in partial fulfillment of the requirements for the degree of

Master of Science in Engineering and Management

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2016 [February 2016]

Signature redacted

Author .....
Systems Design and Management Program
January 15, 2016

Signature redacted

Certified by .....
Patrick Hale
Director, Systems Design and Management Program
Thesis Supervisor

Signature redacted

Accepted by .....
Patrick Hale
Director, Systems Design and Management Program

Latent State Space Models for Prediction

by

Prashan Wanigasekara

Submitted to the Systems Design and Management Program
on January 15, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering and Management

Abstract

In this thesis, I explore a novel algorithm to model the joint behavior of multiple correlated signals. Our chosen example is the ECG (Electrocardiogram) and ABP (Arterial Blood Pressure) signals from patients in the ICU (Intensive Care Unit). I then use the generated models to predict blood pressure levels of ICU patients based on their historical ECG and ABP signals. The algorithm used is a variant of a Hidden Markov model. The new extension is termed as the Latent State Space Copula Model. In the novel Latent State Space Copula Modelthe ECG, ABP signals are considered to be correlated and are modeled using a bivariate Gaussian copula with Weibull marginals generated by a hidden state. We assume that there are hidden patient "states" that transition from one hidden state to another driving a joint ECG-ABP behavior. We estimate the parameters of the model using a novel Gibbs sampling approach. Using this model, we generate predictors that are the state probabilities at any given time step and use them to predict a patient's future health condition. The predictions made by the model are binary and detects whether the Mean arterial pressure(MAP) is going to be above or below a certain threshold at a future time step. Towards the end of the thesis I do a comparison between the new Latent State Space Copula Model and a state of the art Classical Discrete HMM. The Latent State Space Copula Model achieves an Area Under the ROC (AUROC) curve of .7917 for 5 states while the Classical Discrete HMM achieves an AUROC of .7609 for 5 states.

Thesis Supervisor: Patrick Hale
Title: Director, Systems Design and Management Program

Acknowledgments

I would like to thank Kalyan Veeramachaneni and Alfredo Cuesta-Infante for being great mentors for this thesis. They were the guiding forces behind this research. I would also like to thank the members of the ALFA (Anyscale Learning for All) Group for their support throughout my stay.

I would also like to thank Patrick Hale who is now the main supervisor of this thesis for his advice and guidance.

Finally I would like to thank my family and a special thanks to my sister Nirandika Wanigasekara for being there for me and being supportive. This thesis would not have been possible without the help and emotional support from all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In a machine learning and data analytics setting researchers deal with multiple correlated signals (features, variables) from data sets on a day to day basis. This can be in the form of a financial analyst modelling correlated stock prices, or a health care data analyst modelling correlated patient signals, or a bio-engineer modelling correlated DNA sequence structures or in many other forms. In any context modelling the joint behavior of multiple signals and making predictions based off them can be a challenging task. In this thesis we try to address the challenge of modelling the joint behavior of multiple correlated signals and making predictions. We also provide a general framework where the modelling and prediction task can be easily parametrized and used on data from any domain. For this thesis we chose health care as our domain and ECG (Electrocardiogram), ABP (Arterial Blood Pressure) signals from patients in the ICU as our correlated signal pair. This chapter presents the objectives of the research, motivations behind why joint modelling is important, discusses related work, technical challenges and lists the contributions.

1.1 Objectives and Motivations

In this thesis the main objectives and their underlying motivation or rationale are

1. **Modelling joint behavior**

   Modelling the joint behavior of multiple correlated signals is the main objective of this thesis. This raises the natural question of why joint modelling is important.

The underlying rationale is that when signals are correlated we can use information from correlated signals to make better inference and prediction. The information from multiple signals can also be used to reduce the noise, improve signal quality, reduce false alarms [Zong et al., 2004]. If one signal dominates all others joint modelling may only add slight incremental value. But if most signals are correlated but non dominated each new signal can add significant inference and predictive power.

2. **Keeping the original form of the signal intact during the modelling stage.**
   The actual (or at most aggregated) continuous signals are used as input to our algorithm as opposed to other methods where the signals are discretized during the modelling stage. During the discretization process important information related to the signal might be lost whereas our model preserves the original version of the signal. By preserving the original form of the signal we are able to use every minute form of information that is present in the signal for inference and prediction.

3. **Making predictions based on generated models**
   After introducing a joint modelling framework we introduce the associated prediction framework where our models can be used to predict the value of a future outcome and create value to the user. The underlying hypothesis is that joint modelling may lead to better predictive power.

4. **Providing a framework that can be parametrized**
   We hope to provide a framework which can be used in any domain, on any data set for any number of correlated signals. This is achieved by having easily configurable parameters related to the modelling and prediction tasks.

The model that we propose is termed the Latent State Space Copula Model. It is a variant of a Hidden Markov Model (HMM). We assume that there are "latent" or hidden patient "states" that transition from one hidden state to another driving joint ECG-ABP behavior. Some of the reasons why we chose a "latent" state space model are that such models

1. Capture the temporal nature of the signals well

2. Automatically group similar patterns in the patient data into associated hidden states

3. Provide a framework where the prediction parameters are easily configurable.

An in depth discussion about why we chose a latent space model is found in Chapter 3.

In this thesis we use ECG and ABP as our pair of correlated signals. The ECG signal is related to the electrical depolarization and re-polarization of the heart's atria and ventricles. The ABP is more related to mechanical dynamics of the heart (i.e contracting cardiac muscles). Although the signals are in different domains (mechanical vs electrical) they are strictly coupled (since they are generated by the same source) and the temporal, morphological dynamics of both signals and the coupling can be modelled by a single model with different parameters or states. They are very similar in terms of rhythm and timing. There are other signals (which are not used) that are correlated with the ABP signal of an ICU patient. Potential candidates for correlated signals are other vital signals like arterial oxygen saturation, respiration rate, temperature and other data sources such as electronic medical records. By using information from these correlated signals we can build a more comprehensive modelling and prediction structure. Better modelling and prediction structures lead to early detection of critical events and rapid interventions by the hospital staff. This can help improve the patient's condition before it deteriorates and reduce the morality rate.

Moving on to prediction, our specific prediction task in this thesis is to predict the Arterial Blood Pressure (ABP) in an ICU setting. A single Acute Hypotensive Episode (AHE) which is a significant drop in blood pressure is correlated with more than double the risk of death in the ICU among patients in the MIMIC2 data set [Clifford et al., 2009]. "An episode of acute hypo-tension is defined as a prolonged period of time (around 30 minutes or more) during which the mean arterial blood pressure falls and stays below 60 mmHg" [Stell et al., 2009], [Singh et al., 2010][1].The blood pressure fluctuation in the proximity of an Acute Hypotensive Episode (AHE) is shown in the Figure 2-5. The prediction framework will be discussed in Chapter 3

[1]in this thesis we effectively take the time period to be 20 minutes

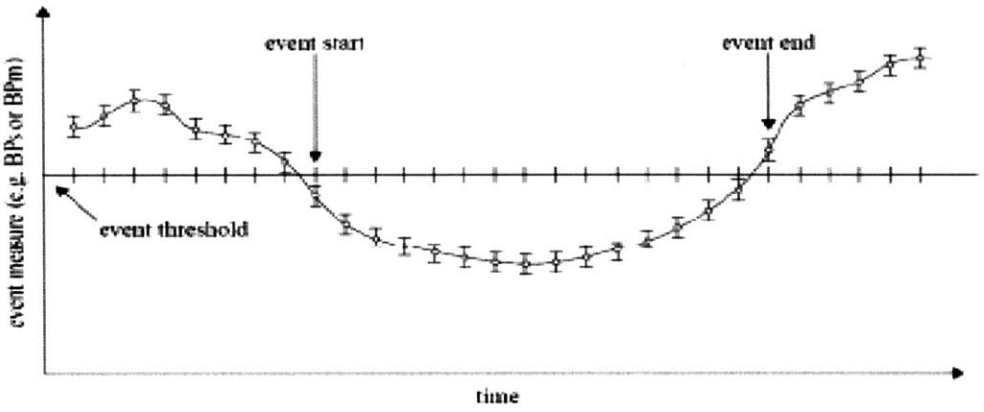


Figure 1-1: Definition of a AHE event.Figure adopted from [Stell et al., 2009]

When modelling the joint behavior of ECG, ABP and making predictions most studies impose assumptions like treating the signals as being independent or assume that the observations follow known distributions. Some of the techniques that are currently used are

1. Gaussian mixture models [Sayadi and Shamsollahi, 2013]
   These models assume that the ECG, ABP signals follow known distributions like the Gaussian distribution. In our approach we try not to impose such restrictions on the ECG, ABP signals and try to further model their joint behavior without making any assumptions on the type of underlying distribution. The correlations are allowed to have non linear characteristics and our model is built to capture that behavior.

2. Dynamic Bayesian Networks [Kim et al., 2014], [Waldin, 2013]
   In these models the observed ABP value is discretized. In some models the observations are considered to be independent. Our model does not discretize the original signal and uses it in its original (maybe aggregated) continuous form. Our model also does not make the independence assumption.

These various modelling techniques will be discussed in depth in section 1.2.

1.2 Related work

As mentioned briefly in section 1.1 there are approaches based on Gaussian Mixture Models, Dynamic Bayesian Networks and a number of other methods to model and predict blood pressure and ECG. Below is a summary of each approach.

1.2.1 Other approaches for joint modelling

- Gaussian Mixture models

  In [Sayadi and Shamsollahi, 2013] a Gaussian mixture model based approach is used to model the dynamics between the ECG, ABP signals. A set of time-varying differential motion equations was used to model the ECG, ABP waveforms. The waveform parameters have a Gaussian Mixture Model representation with parameters for the amplitude, phase (angular speed, angular difference) and noise of the signal. Additionally the joint behavior of ECG, ABP is captured by using the fact that there is phase shift (that dynamically changes) between the ECG and ABP signal and modelling it with an additional set of phase related parameters.
  This model imposes an assumption that the parameters are distributed according to a Guassian distribution. In actuality ECG, ABP distributions may not always display Gaussian behavior. In our model we do not make any such assumptions. The patient data set size in [Sayadi and Shamsollahi, 2013] is 76 patients from the MIMIC database whereas ours is a 650 patient data set.

- Dynamic Bayesian Networks (DBN) and Temporal Bayesian Networks

  In [Kim et al., 2014] a DBN is used to model the ABP waveforms and there is a comparison of the prediction performance between the DBN and K nearest neighbors method. For the Dynamic Bayesian Network first a HMM is trained and the initial, state transition and observation probabilities are obtained. The expectation maximization algorithm is used in the learning process. The data set is divided into 2 sets where one is with patients that had a AHE condition and the other with patients with no AHE conditions. Each data set is used to train a separate model. The posterior probabilities

of a test object is obtained on each model to make a decision on its classification. This method only uses the ABP signal to train the Dynamic Bayesian Network. Our method uses both the ABP and ECG signal for modelling and prediction.

In [Waldin, 2013] a Temporal Bayesian Network is used on the ABP signal to make predictions. An aggregate window is defined and seven features are extracted based on the ABP signal. These features are quantized into 10 discrete bins. These are then used in a Temporal Bayesian Network to perform predictions. The model also assumes that when the state is given the observations are independent of each other.
In our model we do not make this independence assumption and present a structure that can model the correlation between the observations. Additionally we do not discretize the aggregated features and directly use the continuous signal.

Various other methods are also used to model ECG, ABP too. For example in [El-Gohary and McNames, 2007] cross correlation analysis is performed to learn the structure between 2 signals. In [Fong et al.,] echo state networks which is a specific type of neural network is used to model the joint structure of ECG, ABP. There are also attempts to create online realtime systems that use ECG, ABP signals to predict AHE as described in [Stell, 2015].

1.2.2 Previous research

The current research work builds on the work in PhysioMiner:[Gopal, 2014] where the ECG signals were extracted on a beat by beat basis and stored on the Amazon cloud. This ECG beat data for 650 patients was downloaded and used for this thesis. It also builds on work done on ABP beat processing in beatDB:[Dernoncourt et al., 2013]. beatDB is a large scale machine learning and analytics framework for mining knowledge from high resolution physiological waveforms. The ABP beat data for 650 patients were obtained from the beatDB database. There is also prior work on making predictions using only the ABP signal in [Waldin, 2013] where a Temporal Bayesian Network was used. The above mentioned research work were a good foundation and starting point for the research conducted in this

thesis.

1.3 Technical Challenges

The technical challenges that were encountered in this thesis are given below

- **Proposing the extension to the HMM to model the joint ECG, ABP behaviour.**
  In general working on the Latent State Space Copula Model and the novel sampling process involved learning about Copulas, Sklar's Theorem, Weibull distributions and Gibbs sampling (these are discussed in depth in Chapter 3). This was indeed challenging and required a steep learning curve. To the best of the author's knowledge there are no other prior research work that attempts to model correlated variables using HMMs, Copulas and Gibbs sampling in this manner.

- **Processing a large number of patients in parallel for each time step.**
  During each time step of the HMM around 500-650 patients data was required to be processed in parallel. This required a lot of computation power and the processing was distributed over a large number of cores in a distributed computer cluster.

- **Aligning the ECG and ABP signals.**
  Once the ECG, ABP beat data were obtained an aligning algorithm was required to be written to match the ECG, ABP signals.

1.4 Contributions

The research done in this thesis builds on the previous projects mentioned in section 1.2.2 and the new contribution comes in the form of

1. **Aligning and matching the separate ECG and ABP signals.**
   To the best of the author's knowledge there were no other off the shelf algorithms that

extract ECG, ABP signals and sync them by aligning the signals at a beat level. This thesis proposed the matching algorithm while leveraging on the original ECG beat data provided by PhysioMiner:[Gopal, 2014] and the original ABP beat data provided by beatDB:[Dernoncourt et al., 2013].

2. **Proposing the novel Latent State Space Copula Model for jointly modelling the ECG and ABP.**

   The proposed extension to the HMM is novel and is one of the contributions of this thesis.

3. **Comparing the novel Latent State Space Copula Model with a Classical Discrete HMM.**

   The proposed novel structure is compared with the industry standard so that the reader can get an idea of the performance and prediction accuracy.

1.5 Organization

An overview of the data set, waveforms, feature extraction and the challenges faced while syncing and aligning the ECG, ABP signals is discussed in Chapter 2. An introduction to the novel Latent State Space Copula Model, HMMs and Copulas is given in Chapter 3. The experiment setup and challenges in data processing are discussed in Chapter 4. The results are discussed in Chapter 5 where the Latent State Space Copula Model has a AUROC of .79175 for 5 states. Conclusion and future work is given in Chapter 6.

Chapter 2

Data Processing and Alignment

This chapter will first give an overview of the MIMIC II database. Then the individual ECG, ABP waveforms will be presented along with a figure depicting their tightly coupled nature. We hope that this will give the reader an idea of the data set we are using and the nature of signals that we are trying to model. Then we will discuss challenges faced while pre-processing the raw ECG, ABP signals. Although the pre-processing step is not a direct contribution of this thesis it is included because it's a vital part of the overall process. A section about feature extraction will follow. Towards the end, the distributions for the sub selected 650 patient data set that were used for the experiments will be shown and the chapter will conclude with the signal aligning algorithm which is a direct contribution of this thesis.

2.1 Overview of the MIMIC II data set

Our data is taken from the Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC II) version 3 waveform database. The data is recorded from patients at Boston's Beth Israel Deaconess Medical Center over a 7 year period. The total database has records for 25,328 ICU stays from 22,870 hospital admissions. "Of those admissions, 1360 (5.9%) had multiple ICU stays, and on average there were 1.11 ICU stays per hospitalization. The median ICU stay lasted 2.2 days" [Saeed et al., 2011], [Clifford et al., 2009].
There are 2 main types of data in the MIMIC II database

1. Clinical Data - These are data from patients' electronic medical records, doctors notes collected form the hospital medical systems etc.

2. High resolution physiological data - These are data captured from bedside monitor waveforms, associated derived parameters and events.

The ECG, ABP physiological signals that are used in our study were captured by Patient monitors (Component Monitoring System Intellivue MP-70; Philips Healthcare) that were located at the bedside of ICU patients. The patient bedside monitor acquires, digitizes, process the signals and produces alarms when critical events occur [Saeed et al., 2011].

The waveform Database is an aggregate of 23,180 sets of recordings which amount to over 3 TB of data [Saeed et al., 2011], [Waldin, 2013]. Ideally one recording is supposed to map to one patient except in some rare instances where there are recording anomalies. The time of the recordings can range from a few hours to a few days (the average duration being 2.2 days) [Saeed et al., 2011]. Out of all the patients in the MIMIC II v3 database, 6232 patients have ABP waveform records [Dernoncourt et al., 2013], [Waldin, 2013], [Kim et al., 2014] [Saeed et al., 2011]. The ABP measurements are from the arterial lines and the lines generally placed in the same location for each patient. The ECG data used in this research was obtained form [Gopal, 2014] where the ECG line II signal from 6118 patients from the MIMIC database was processed. From the 6232 patients with ABP readings and 6118 patients with ECG readings, 6118 patients had both ECG and ABP readings. Out of those 6118 patients 650 patients were randomly selected for our study. We limited ourselves to 650 patients because our main objective was not to show the scalability of our algorithm but to show how well we can model the joint behavior of ECG, ABP. 650 patients is a moderately sized data set when compared to studies like [Sayadi and Shamsollahi, 2013] where only 76 patients were used, [Fong et al.,] where 6 patients were used, [Li and Clifford, 2012] where data from 182 visits were used. On the other end of the spectrum [Waldin, 2013] uses 1000 patients, [Dernoncourt et al.,] uses 5000 patients. We consider our 650 patient data set to be of appropriate size to highlight the challenges of using the Latent State Space Copula Model while not being hampered by scalability and computation issues. The associated ECG, ABP distributions for 650 patients will be given in section 2.5.

2.2 The individual ECG, ABP waveforms

In this section we take a look at the individual ECG, ABP waveforms and the strict coupling that exists between them in order to get a better idea about the signals that we are modelling. This section gives a very brief overview of the waveforms and sets up the vocabulary which will be helpful when discussing challenges in the next section.

An isolated ABP beat waveform is given in Figure 2-5.

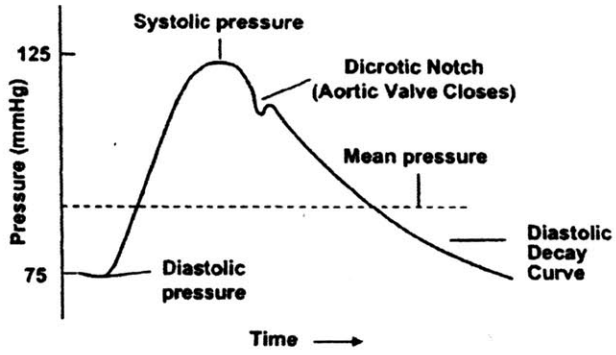


Figure 2-1: Chracteristic ABP waveform.Figure taken from [Cornell, 2011]

In Figure 2-5 the Systolic pressure (P_s) is the highest pressure reached and this is when the heart contracts and sends blood out. The Diastolic pressure (P_d) is the lowest pressure reached and this is when the heart refills with blood. The Dicrotic Notch is a short dip in the blood pressure due to a back flow of blood [PhysiologyWeb, 2015]. The mean arterial pressure is defined as $\frac{2P_d+P_s}{3}$.

The characteristic ECG beat waveform is shown in Figure 2-2.

The ECG signals in Figure 2-2 capture waveform components which indicate electrical activities during one heart beat. These waveforms are divided into a P wave, QRS complex and T wave. The P wave is associated with the atria contracting and pumping blood into the ventricles. The QRS complex, which generally starts with a small downward dip Q; a larger upward peak R; and then a downwards S wave is associated with ventricular depolarization and contraction. The T wave is a small upwards waveform representing ventricular repolarization. The PR interval is an indication of the transit time for the electrical signal to travel from the sinus node to the ventricles [Dubin, 2000].

The tight coupling that exists between the ECG and ABP signal is shown in Figure 2-3.

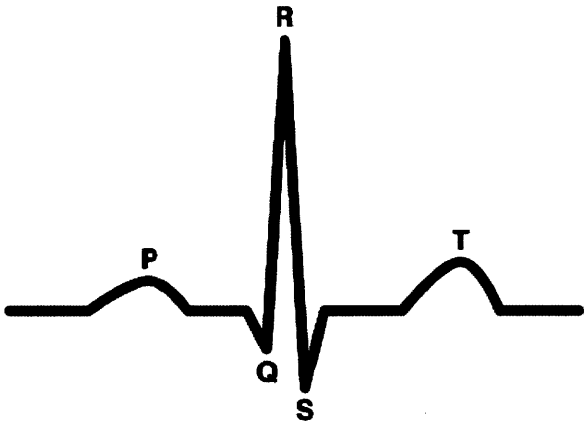


Figure 2-2: Chracteristic ECG waveform.

We see that the beginning of the ABP systole happens some time period (180 ms in Figure 2-3) after the R wave of ECG. "The interval between the R wave and the upstroke of the systole represents the delay between actual ventricular depolarization and the arrival of the signal to the pressure transducer" [Yartsev, 2015]. This delay between the ECG, ABP can vary from patient to patient and within a single patient based on the patients health condition.

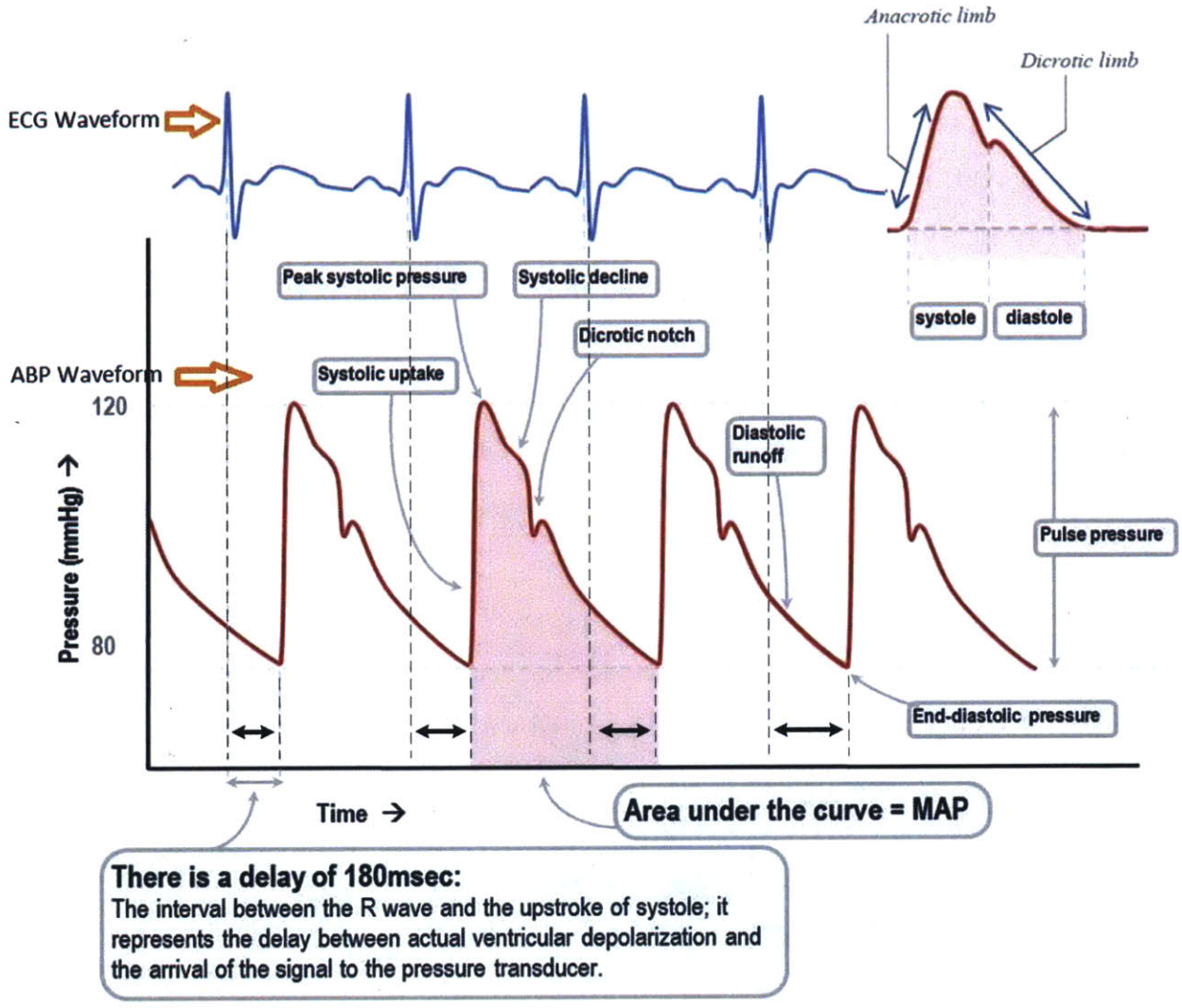


Figure 2-3: The coupling between the ECG and ABP waveforms. The phase shift that exists between the ECG, ABP signal is highlighted in this figure. Figure taken from [Yartsev, 2015]

2.3 Challenges faced while processing raw ECG, ABP signals

This section will discuss the challenges faced while bringing the raw ECG, ABP signals to a state that can be given to a machine learning algorithm. As stated in Chapter 1, one of the main contribution of this thesis is the signal aligning algorithm (discussed in section 2.6) and not the actual processing, cleaning or validation of the raw ECG, ABP waveforms. Nevertheless the pre-processing of signals is a vital component of the overall process and this section will provide a brief overview. These challenges were actually addressed and implemented in [Gopal, 2014] and [Dernoncourt, 2014].

Our data is recorded in an ICU setting which can be chaotic at times (i.e the patients having sudden movements). This contributes to a large number of misaligned and invalid signals. The raw signals are also non-contiguous. By non-contiguous we mean that the time series of the ECG, ABP signal can be broken at certain points due to causes like readjustment of the measuring devices. When this happens we identify such occurrences and divide the signal into contiguous sections called "segments".

One of the key insights in pre-processing is to extract the ECG, ABP signals at the beat level. We hypothesize that the key features that lead to modelling and prediction should be captured at the beat level. The features associated with a beat can change from patient to patient and within a single patient as the patient's health condition changes.

As part of overcoming the challenges mentioned above the following steps were taken (in order) to pre-process the ABP signals :

1. To detect the ABP beats an ABP beat onset detection algorithm described in [Zong et al., 2003] was used. This returns the start, stop indexes of the beat.

2. The validy of the beats were identified using the algorithm described in [Sun et al., 2006]. ABP Beats that do not pass a certain threshold were removed.

3. Beats that are contiguous were identified and grouped. Whenever two consecutive beats are more than 5.6 seconds apart, they are considered to be non-contiguous

[Waldin, 2013]. A contiguous series of ABP beat measurements is termed as a segment.

As part of pre-processing the ECG signals the following steps were taken in order :

1. A QRS detection (the QRS complex was introduced in section 2.2) algorithm was run on the ECG signal to detect the QRS complexes. For this the QRS detection algorithm introduced in [Zong, 1998] was used. The algorithm annotates the beginning and end of a QRS complex.

2. The signal was divided into beats around the identified QRS complexes. The beat division algorithm started at the left side of a QRS complex and travelled in either direction appropriately to detect beat divisions [Gopal, 2014].

One of the challenges faced during ECG signal pre-processing that was specially mentioned was that the QRS detector mistakenly detected P waves to be QRS waves. This was overcome by subdividing the signals and running a QRS detector on the subdivided chunks [Gopal, 2014].

By going through these pre-processing steps [Gopal, 2014] and [Dernoncourt, 2014] were successfully able to segment the ECG, ABP signals at the beat level. These initial efforts were vital for the research conducted in this thesis. This concludes the brief overview of the pre-processing stage. A discussion about feature extraction, the sub-sampled 650 patient data set and the aligning algorithm will follow.

2.4 Extracting features from the ECG, ABP signals

After pre-processing the data we are ready to extract features for the ECG, ABP signals at the beat level. For ABP we first extract the mean arterial pressure $\frac{2P_d+P_s}{3}$ for every valid beat in a "segment" (a contiguous time interval which has an uninterrupted signal). Later on we would define an aggregation interval and get summary statistics like the mean, standard deviation, kurtosis and skew associated with this interval [Waldin, 2013]. In our study the mean of the mean arterial pressure (MAP) was used as a feature. For ECG, again contiguous

time segments were identified and features like the ECG heart rate, Q-S interval length, Q-R interval length and wavelet features were extracted for each beat. Later on we would define an aggregation interval and get summary statistics for ECG associated with the aggregation interval. In our study we used the mean of the heart rate as a feature.
After describing the feature extraction step we would now move on to looking at the distributions of the chosen ECG, ABP feature values for the 650 patient data set.

2.5 The ECG, ABP distributions for the chosen data set

The ECG heart rate distribution and ABP mean arterial pressure (MAP) distribution for the randomly chosen 650 patients in our data set is shown in Figure 2-4, Figure 2-5. The following scale is used for the x axis of the plots

- ECG : $[0, 1] \rightarrow$ [0.0082 , 359.63] beats per minute
- ABP : $[0, 1] \rightarrow$ [31.81 , 167.39] mmHg

The y axis shows the ECG, ABP counts. The distributions shown in Figure 2-4, Figure 2-5 are for our 650 patient population and they are the true underlying distributions that we are trying to model. In the context of inference, modelling these distributions are important because once we have an accurate generative model we can use it to "generate" future patient ECG, ABP signal values.

We are interested in predicting incidents where the ABP is below 60 mmHg. In the 650 patient data set there are only 6.0344% of instances where the ABP is below 60mmHg for 20 minutes. And there are only 3.25% of instances where ABP is below 60 mmHg for 40 minutes. Thus it is very sparse in the incidents that we want to predict. This is usually the case with many real world data sets and to make predictions based on sparse data the prediction algorithms have to be very robust. These statistics are highlighted in 2-6 where a histogram of the number of Acute Hypotensive Events per patient is shown.

After getting a rough idea about the data set we move on to explain the ECG, ABP matching algorithm which is one of the contributions of this thesis.

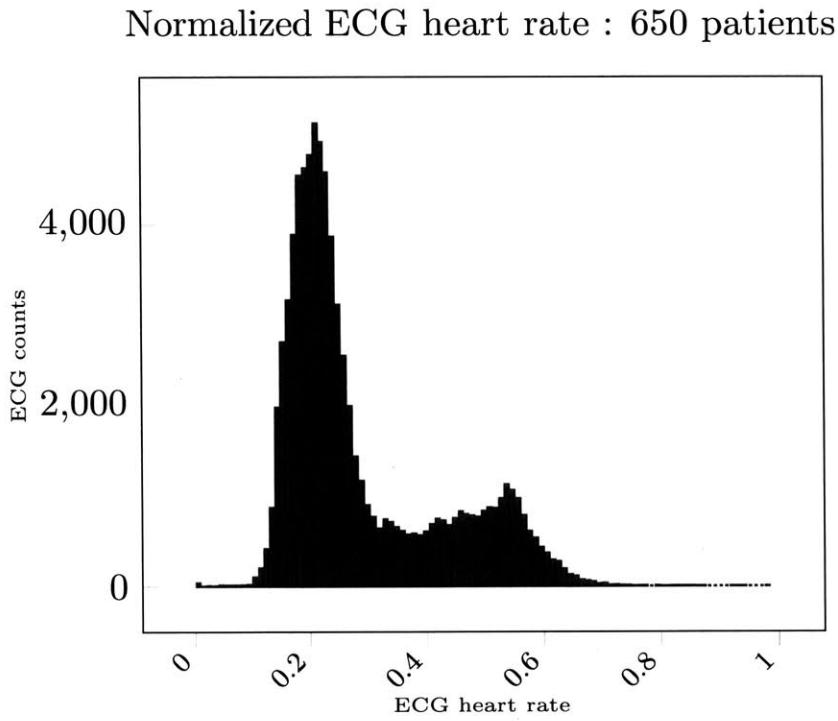


Figure 2-4: Distribution of the ECG heart rate signals for 650 patient

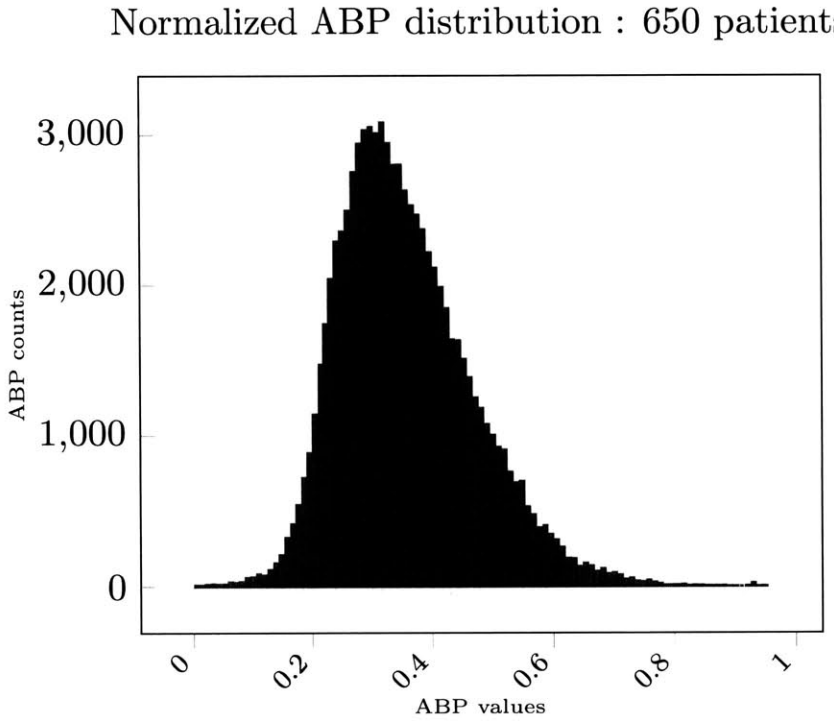


Figure 2-5: Distribution of the ABP mean arterial pressure for 650 patient

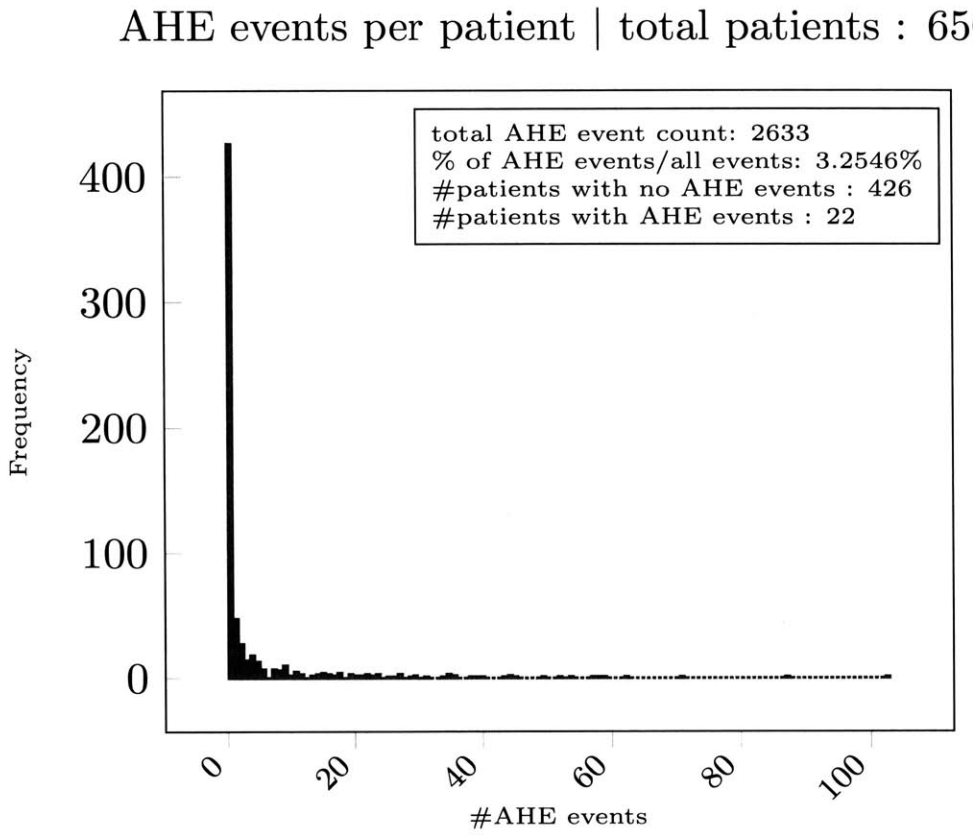


Figure 2-6: Histogram of AHE events per patient

2.6 Aligning the ECG, ABP waveforms

As part of the contribution of this thesis we take the 2 standalone ECG and ABP signals and align them so that their time intervals match. We then perform an aggregation over an aggregation window of 150,000 measured ECG & ABP sample indices (which corresponds to a 20 min interval $= \frac{150{,}000}{125*60}$, assuming a 125Hz sampling rate) and obtain the average ABP and ECG values for the aggregation window. The data assembly algorithm is described in this section and our end result is a 3-tuple shown in Table 2.1. The interpretation of the columns in Table 2.1 is given below. Note that the ECG, ABP share the same sample index and this is helpful when aligning the signals.

- **Sample index start** - The ECG, ABP is sampled at 125 Hz. This column shows the sample index at the start of the aggregation window of size 150,000 sample indices. The ECG, ABP share the same sample index and this is helpful when aligning the

signals.

- **ECG heart rate** - The Aggregated heart rate over the specified interval.
- **ABP** - The aggregated ABP over the specified interval

(Sample Index start, aggregated heart rate, aggregated ABP)

Sample index start (of the aggregation window)	ECG Heartrate (beats per min)	ABP (mmHg)
1526565	114.26	105.16
1676565	113.83	122.90
1826565	114.35	122.15
1976565	113.80	120.00

Table 2.1: Final aggregated and aligned ECG, ABP table or patient ID 3003618

Note that that there is at least a 150,000 sample index gap between each adjacent entry in Table 2.1. The aligning algorithm is presented next with an example for the ECG, ABP tables for patient ID 3003618.

The ECG table for patient ID 3003618 is shown in Table 2.2. The interpretation of the columns are given below

- **Sample index** - The ECG is sampled at 125 Hz. The sample index is the sampling index at the beginning of the ECG beat. The ECG, ABP share the same sample index and this is helpful when aligning the signals.
- **ECG Heart rate** - The measured ECG Heart rate value.

Sample Index (index)	ECG Heartrate (beats per min)
1526565	115.38462
1526631	115.38462
1526696	113.63636
1526762	113.63636

1526826	115.38462
...	...

Table 2.2: ECG beats file for patient ID 3003618

The ABP table for patient ID 3003618 is shown in Table 2.3. The definitions for the columns are given below

- **Sample index start** - The ABP is sampled at 125 Hz. The sample index is the sample index at the beginning of the ABP beat. The ECG, ABP share the same sample index and this is helpful when aligning the signals.
- **Sample index end** - The sample index is the sample index at the end of the ABP beat. The ECG, ABP share the same sample index and this is helpful when aligning the signals.
- **Valid status** - This indicates whether the ABP beat is valid.
- **ABP** - The mean arterial pressure of the beat in mmHg.

sample index start (index)	sample index end (index)	valid status (binary)	ABP (mmHg)
1528198	1528243	1	71.1924075409
1528243	1528264	0	nan
1528264	1528309	0	nan
1528309	1528341	0	nan
1528341	1528396	0	nan
1528463	1528528	1	119.500527267
1528528	1528594	1	124.797983314
1528594	1528659	1	118.674359136
...	...	...	...

Table 2.3: ABP beats file for patient ID 3003618

Signal aligning algorithm

The following steps were taken when aligning the ECG and ABP beats

- Step 1: Select a window length w. (i.e $w = 150,000 \rightarrow 20min$)
- Step 2: Get the starting point for the aggregation window. An immediately available, valid entry is read from the *sample index* column of the ECG table (Table 2.2). Note that there is no specific reason for choosing ECG over ABP for this step.

$$sample\ index\ start = \ sample\ index\ of(ECG)$$

(i.e *sample index start* = 1526565 read from Table2.2)

- Step 3: Compute the end point of the aggregation window

$$sample\ index\ end = sample\ index\ start + w$$

(i.e *sample index end* = 1526565 + 150000 = 1676565)

- Step 4: Compute the average of the ECG, ABP in the selected range

$$range = (sample\ index\ start, sample\ index\ end).$$

i.e $range = (1526565, 1676565)$

- Step 5: Slide the window a length of w indexes forward
- Step 6: Go back to step 2 and repeat the process until the end of either the ECG or the ABP file is reached.

Some of the edge cases that we had to consider were

1. In step 4: What if the exact sample end index (i.e 1676565 in the example) couldn't be found in the corresponding ECG, ABP signal of the patient?
   Even if an exact index match is not found we still aggregate the ECG, ABP signals that fall within that range (i.e $range = (1526565, 1676565)$. This also does not affect the next range which is $range\_next = (1676565+150000, 1676565+150000*2))$. If we encounter time periods where 40% of the indexes within a 150,000 interval is missing then we consider the patient's contiguous time series signal to end at that point in time.

2. In step 4: What if there are no ABP readings for the corresponding time period (i.e $range = (1526565, 1676565)$)?

   If there are no ABP readings in the corresponding ECG time period we consider the time series signal to end at that point. Basically we stop when we encounter a time period that has missing data for either ECG or ABP. This prevents us from having time series signals that have missing 20 minute blocks (our aggregation time interval is a 20 minute block) for ECG or ABP observations.

3. How many NaN's will be tolerated within a time interval before considering the time period as invalid?

   If there are more than 40% NaNs the time step is considered as invalid and we end the time series signal at that time step.

4. If there is a levy allowed in the end point of the time period, how would we deal with a stacking up effect?

   In general we average ECG, ABP beats in a 150,000 time interval which is around 20 minutes. We start at a point and keep moving the 150,000 window forward. For example if we started at time index t_idx the time intervals will be within $\{t\_idx,\ t\_idx+ 150,000,\ t\_idx + 150,000 * 2, ..\}$. Because of this some time intervals may have more beats than others but this does not lead to a stacking up effect. Since we average over a 20 minute interval, make sure that at least 60% of the indexes have valid beats and make sure to stop if more than 40% of the beats are missing or NaNs, we get accurate alignment and aggregation.

The final output is shown in Table 2.1 at the beginning of this section. The aligning algorithm performed well because we leveraged the ECG, ABP beats that were already processed at the beat level in [Dernoncourt, 2014] and [Gopal, 2014]. After aligning the ECG, ABP signals we are ready to perform the modelling and prediction. The subsequent Chapters will discuss these efforts in detail.

Chapter 3

Latent State Space Models

This chapter discusses the modelling stage of this thesis. First we discuss why we chose Hidden Markov Models. Then we show that we have 2 different approaches when using HMMs. One being the Classical Discrete HMM which has been around for decades. The second approach is the Latent State Space Copula Model which is a novel method. The Latent State Space Copula Model was conceived by Kalyan Veeramachaneni and Alfredo Cuesta-Infante who, as mentioned in the acknowledgement section are the guiding forces behind this thesis. Each of the 2 approaches have a dedicated section. A section on Model Prediction then follows where we discuss how the Latent State Space Copula Model is used in a prediction context.

3.1 Why Hidden Markov Models

We chose Hidden Markov Models because we wanted to capture the temporal trends in a signal for classification, by modeling the temporal information as a sequence of "state" changes. HMMs are well known for modelling the connections between adjacent states, symbols, sequences and is heavily used in speech recognition [Rabiner, 1989], digital communications and biological sequencing [Yoon, 2009]. In general, a patient becoming hypotensive is not only dependent on the present vital sign measurements, but also on a sequence of measurements from the past. These relationships can be modelled well with HMMs. Additionally the notion of "states" allow us to automatically group similar patterns or similar patient

signals together (i.e a state that's related to low ABP, low ECG vs low ABP, normal ECG). This leads to better predictions. A HMM structure also allow us to model different lead and lag times for a prediction problem. An in depth discussion about the prediction problem will be presented in section 3.6. To be consistent this chapter will use the notation given below

t - time

X - states in a hidden markov model

X_t - state at time t

N_x - total number of states

Y_t - ECG observation at time t

Z_t - ABP observation at time t

A HMM operates on the Markovian assumption which is

1. Given the state X_{t-1} at time $t-1$ the state X_t at time t is independent of the past

$$P(X_t = i|X_0^{t-1}) = P(X_t = i|X_{t-1})$$

Where $i \in \{1, 2, ..., |N_x|\}$

2. The observations or emissions only depend on the current hidden state

3.2 Two different approaches

In this thesis we evaluate 2 different approaches for blood pressure prediction. The first approach is to use a Classical Discrete HMM. This is a standard algorithm that is widely used in the research community. When using the Classical Discrete HMM in addition to the Markovian assumptions there are other assumptions made. It is generally assumed that the observations (ECG, ABP in our case) are independent of each other and that those observations are observed in discrete values.

The second approach which is the Latent State Space Copula Model throws away these 2 additional assumptions mentioned above and tries to view the observations (ECG, ABP in our case) as correlated, continuous variables. At the start of this research we do not exactly

know whether this new approach would improve prediction accuracy. The thesis is in a way an effort to evaluate how the Latent State Space Copula Model compares with the Discrete HMM. Our current patient data set size is 650 patients. The prediction accuracy of specific algorithms is shown to vary with the size of the data set as noted in [Dernoncourt, 2014]. This prompts us to state the data set size whenever we make a statements about prediction accuracy.

3.3 Classical Discrete HMM

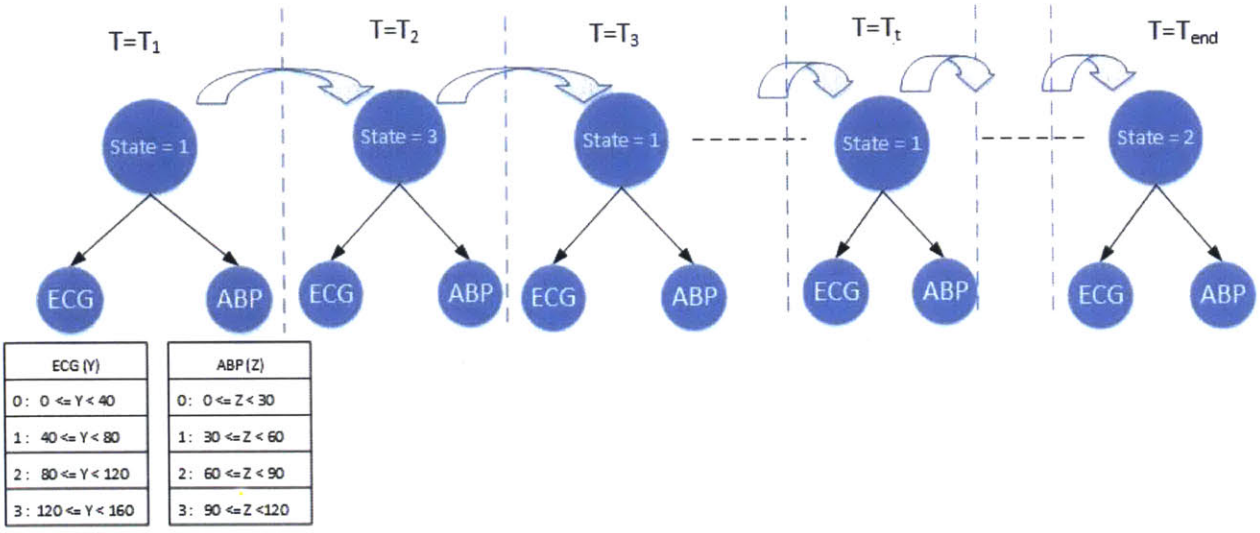


Figure 3-1: Classical Discrete HMM

As mentioned above Classical Discrete HMMs are widely used in the research community. The structure of the Classical Discrete HMM related ECG, ABP is given in Figure 3-1. The time series in the Figure 3-1 is a sequence from $T = \{T_1, T_2.., T_t, ...T_{end}\}$. The ECG, ABP observations have discrete values ranging from $\{0, 1, 2, 3\}$ which correspond to ranges of the original signal.

A brief overview of a Classical Discrete HMMis given below.

The sources used are

1. A tutorial on hidden Markov models [Rabiner, 1989]

2. Hidden Markov Models for modeling blood pressure data to predict acute hypotension

[Singh et al., 2010]

"Consider a discrete time signal of length T. This signal is first subdivided into L windows, each window consisting of T/L samples" [Singh et al., 2010]. In our case L is a 20 minute aggregation window and we get a time series of T/L samples where each entry is a 20 minute aggregate of the original time series of the ECG, ABP signal as discussed in Chapter 2. We then obtain features related to each time window which is considered as our observation (or emission) sequence $Y = Y_1, Y_2, ..., Y_t$. For our model the observations are the mean of ABP, mean of ECG for the 20 minute window. For each patient we will have a sequence of such observations. Hidden Markov models are widely known to be able to model observation sequences by making the assumption that those observations are driven by "latent" states that switch between each other based on a transition and emission matrix that can be learnt. Figure 3-2 shows a standard 3 state Markov chain where the state numbers are inside the

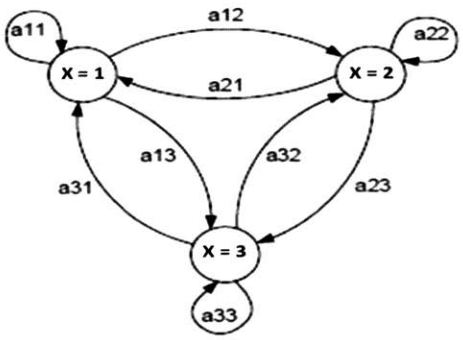


Figure 3-2: A generic 3 state HMM, with state transition probabilities shown on the transition arrows. Figure taken and modified from [Singh et al., 2010].

circles and state transition probabilities are on the arrows. In a more general case consider a system to have N_x distinct states $i \in \{1, 2, ..., |N_x|\}$ and be in one of them at a given time. The transition from one state to another happens at regularly spaced, discrete times based on state transition probabilities associated with the current state. The state transition probabilities are given by

$$a_{ij} = P(X_t = j |\ X_{t-1} = i)\ ;\ 1 \leq i, j \leq |N_x|$$

where X_t is the state at time t and X_{t-1} is the state at time $t - 1$.

The next set of probabilities associated with an HMM are the observation or emission probabilities. While in a state, a state specific emission or observation occurs. The emissions or observations follows a probability distribution associated with that particular state. The purpose of the training phase of an HMM is to learn these transition and emission probabilities. "Using these observation probability distributions and the transition probabilities, it is possible to come up with a state transition sequence $X = X_1, X_2, ...X_t$ for the given HMM, which can "best explain" a given observation sequence $Y = Y_1, Y_2, ..., Y_t$. The parameters that characterize the Classical Discrete HMM are given below:

1. The state transition probability matrix, $A = \{a_{ij}\}$.
2. The probability distribution B of observations Y at each state which will be learnt during the training phase. $B = P(Y|state)$
3. The initial state distribution $\pi = \{\pi(x_1)\}$ (the probability mass function of the initial state distribution).

The following notation is used to describe a Classical Discrete HMM:

$$\Theta = (A, B, \pi)$$

Given an HMM and some observation sequences, there are two types of computations that are relevant for the application at hand:

1. Learning HMM parameters : Given the observation sequence(s) $Y = Y_1, Y_2, ..., Y_t$ find the parameters of the HMM $\Theta = (A, B, \pi)$ that maximizes the probability $P(Y|\Theta)$.
2. Given a model $\Theta = (A, B, \pi)$, and an observation sequence $Y = Y_1, Y_2, ..., Y_t$ compute $P(Y|\Theta)$, that is, the probability that the given model generated the given sequence.

Step 1 is used for training the HMM using a given observation sequence. The Baum-Welch algorithm which uses the expectation maximization algorithm is a well known procedure for training an HMM.[1]

[1]The material related to the generic HMM presented above is referenced form [Singh et al., 2010] and [Rabiner, 1989].

There are a variety of off the shelf Classical Discrete HMM's available. Commercial software like MATLAB, R, python offer HMM functionality in their toolboxes. For this thesis I used the HMM toolbox implemented in MLBlocks [Collazo, 2015]

MLBlocks

MLBlocks [Collazo, 2015] is a machine learning system that enables the user to use Discriminative Modeling, Generative Modeling techniques to make predictions on any data set. It is highly parameterizable. In this thesis MLBlocks is chosen above others (i.e MATLAB, R) to train a Classical Discrete HMM on the 650 patient data set.

3.4 Latent State Space Copula Model

The main contribution of this thesis is to propose this novel structure in order to better model the correlations between the ECG and ABP signals. The model also allows the observations to be continuous thus we also expect it to capture more information. An in depth discussion of the Latent State Space Copula Model is given in the following sections. Instead of diving right into explaining the model, the concepts and distributions that are used in the Latent State Space Copula Model will be introduced first. Thus a section on Weibull distributions, Copulas and Sklar's Theorem will be followed by the section that introduces the Latent State Space Copula Model.

3.4.1 Weibull distribution

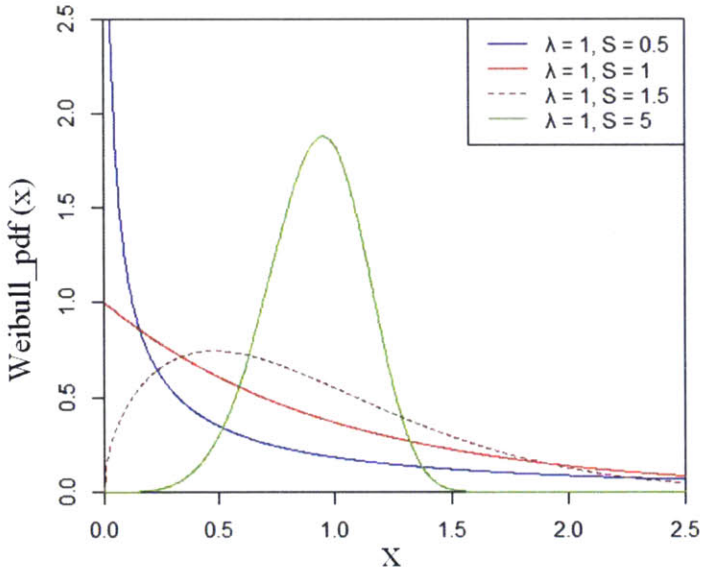


Figure 3-3: PDF of a Weibull distribution with $scale(\lambda)$, $shape(S)$.

The observations (Y, Z) for (ECG, ABP) for time $T = \{T_1, T_2, .., T_t, .., T_{end}\}$ are real valued. We chose to model the ECG, ABP observations individually using Weibull Distributions. The Weibull PDF is a 2 parameter distribution with $scale(\lambda)$, $shape(s)$. It's usually

used to model failure rates. By changing the $shape(s)$ parameter the Weibull distribution can interpolate between the exponential distribution ($shape(s) = 1$) , Gaussian and the Rayleigh distribution. We chose the Weibull dostibution because of this flexibility.

The probability density function of a Weibull random variable with $scale(\lambda), shape(s)$ is

$$f(x;\lambda,s) = \begin{cases} \frac{s}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases}$$

In the above equation x is a generic parameter.

- If shape $s < 1$ (shown in blue)

  The Weibull distribution behaves as shown in blue in the plot.

- If shape $s = 1$ (shown in red)

  The Weibull distribution reduces to an exponential distribution.

- If shape $s > 1$ (shown in the dashed line and green)

  The Weibull distribution can take Gaussian, Rayleigh distribution forms.

Sources for the Weibull distribution: [Abernethy, 1996], [Rinne, 2008], [McCool, 2012], [Liu et al., 2010]. The next important building blocks of the Latent State Space Copula Model are Copulas and Sklar's theorem and they will be explained in the next sections.

3.4.2 Introduction to Copulas and Sklar's Theorem

The word Copula is a Latin noun that means "A link, tie,bond" [Simpson, 1968], [Nelsen, 2007].

Copulas

Copulas are used to model multivariate probability distributions and how multiple random variables are linked. There is a special restriction where the marginal distributions of each random variable has to be uniform. In other words copulas are "functions that join or couple multivariate distribution functions to their one-dimensional marginal distribution functions" [Nelsen, 2007].

Applications with a large number of correlated random variables can be modelled well with Copulas and they can be broken down into separate marginal distributions and a copula. Some of the copula families that are available are Gaussian, Archimedean. The Archimedean copula has subfamilies (i.e Clayton, Frank etc) where there are parameters to control the dependence between variables. Copulas are widely used in quantitative finance, reliability engineering, warranty data analysis, weather research etc.

Mathematical Definition of Copulas

(In this subsection we deviate from the notation table introduced in the beginning of the chapter. Thus consider X to be a generic random variable.)
Consider a random vector $(X_1, X_2, ..., X_d)$ where the marginal cumulative distribution functions (CDF's) are given by $F_i(x) = P(X_i \leq x)$. Consider the vector $(U_1, U_2, ..., U_d) = (F_1(X_1), F_2(X_2), ..., F_d(X_d))$ to have uniformly distributed marginals. Then the copula C of $(X_1, X_2, ..., X_d)$ is defined as the joint CDF of $(U_1, U_2, ..., U_d)$

$$C(u_1, u_2, ..., u_d) = P[U_1 \leq u_1, U_2 \leq u_2, ..., U_d \leq u_d]$$

Here the copula C contains information on the dependence structure between the variables $(X_1, X_2, ..., X_d)$. [Nelsen, 2007], [Dalla Valle, 2009], [Veeramachaneni et al., 2015].

Sklar's Theorem

Sklar's theorem states that a multivariate joint distribution can be completely represented by its univariate marginal distribution functions and a copula that models how the univariate marginal distribution functions are dependent on each other [Nelsen, 2007].

Mathematical Definition of Sklar's Theorem

(In this subsection we deviate from the notation table introduced in the beginning of the chapter. Thus consider X to be a generic random variable.)
Consider a random vector $(X_1, X_2, ..., X_d)$ where the marginal cumulative distribution functions (CDF's) are given by $F_i(x) = P(X_i \leq x)$. Consider a multivariate cumulative distri-

bution function H where

$$H(x_1, x_2, ..., x_d) = P(X_1 \leq x_1, X_2 \leq x_2, ..., X_d \leq x_d)$$

then the joint cululative distribution function can be described by only using the marginals CDF's and a Copula C as shown in [Nelsen, 2007] page 18.

$$H(x_1, x_2, ..., x_d) = C(F_1(X_1), F_2(X_2), ..., F_d(X_d))$$

In case that the multivariate distributions has a density f then the following will hold

$$f(x_1, ..., x_d) = c(F_1(x_1), ..., F_d(x_d)) * f_1(x_1) * ... * f_d(x_d)$$

where

$x_1, ..., x_d$ - are random variables

c - is the density of the copula

f_i - the marginal distribution of x_i

F_i - the cumulative distribution function of x_i

After introducing the Weibull distribution, Copulas and Sklars theorem we will now go into the details of the Latent State Space Copula Model.

3.4.3 The structure of the Latent state space copula model

In this section the Latent State Space Copula Model will be introduced in formal notation as well as using graphics. First the equation related to the multivariate joint ECG, ABP distribution will be presented. Then the structure of the Latent State Space Copula Model will be discussed using Figure 3-4 and Figure 3-5.

The equation that models the joint behavior in the Latent State Space Copula Model is given below in formal notation. Given that we are at time t, the state is $x_t = i$, the random variable related to ECG at time t is y_t, the random variable related to ABP at time t is z_t and using Sklar's theorem the multivariate joint distribution between ECG, ABP can be

written as

$$p(y_t, z_t) = c(F(y_t), F(z_t); \Sigma^{(i)}) * p(y_t; \lambda_y^{(i)}, S_y^{(i)}) * p(z_t, \lambda_z^{(i)}, S_z^{(i)})$$

where

the superscript (i) denotes that the variable is associated with state (i)

c - density of the copula

F - CDF of the random variable

$\lambda_y^{(i)}$ - scale of the Weibull distribution for ECG

$S_y^{(i)}$ - shape of the Weibull distribution for ECG

$\lambda_z^{(i)}$ - scale of the Weibull distribution for ABP

$S_z^{(i)}$ - shape of the Weibull distribution for ABP

$p(y_t; \lambda_y^{(i)}, S_y^{(i)}) = Weibull(y_t; \lambda_y^{(i)}, S_y^{(i)})$

$p(z_t, \lambda_z^{(i)}, S_z^{(i)}) = Weibull(z_t; \lambda_z^{(i)}, S_z^{(i)})$

$\Sigma^{(i)}$ - a parameter that models the dependence structure

The above equation is the underlying equation that uses Copulas, Sklar's theorem and Weibull distributions. It models the much anticipated joint behavior of ECG, ABP. We see that the joint distribution is broken down to the product of marginals that are Weibull distributions and a copula according to Sklar's theorem. We chose our copula to be a Gaussian copula because such copulas (along with Frank and Plackett Copulas) model positive as well as negative dependence well [Wang et al., 2008]. Next, we will graphically convey the ideas related to the Latent State Space Copula Model and hope that the reader will find it more intuitive. The structure of the Latent State Space Copula Model is given in Figure 3-4.

The key points to note are that the ECG, ABP circles are connected with double sided arrows. This indicates that those observations are modelled as being correlated. The ECG, ABP observations are also denoted as continuous random variables $(Y_{ECG},\ Z_{ABP}) \in \mathbb{R}$. Modelling the correlation between 2 continuous correlated observations in a HMM time slice can be a challenging task. The novel structure shown in Figure 3-4 achieves this. In Figure 3-4 for each time slice

1. The patient state can be in 1 out of the $|Nx|$ total number of states. For example if the total number of states $N_x = 3$, then the patient state at time t; X_t can be

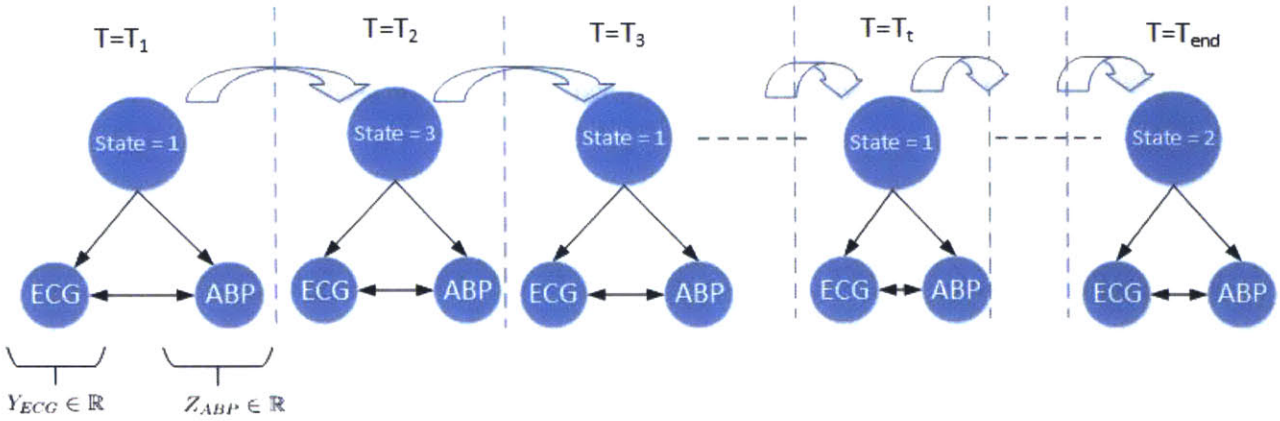


Figure 3-4: The structure of the Latent State Space Copula Model for ECG, ABP observations

$$X_t = i;\ i \in \{1, 2, 3\}.$$

2. For a given state we model the ECG, ABP to be drawn from a Weibull distribution that's specific to that state. If $X_t = i$, then the ECG, ABP are drawn from a Weibull with $scale = \lambda_i,\ shape = S_i$.

3. The values $scale = \lambda_i,\ shape = S_i$ of the Weibull distributions are learnt during the training stage through a Gibbs sampling process discussed in section 3.5.

We zoom into the model further in Figure 3-5 and look at what happens during a single time slice. This will give us a better picture of how the joint pdf is calculated.

The steps involved are given below with explanations for each vertical level in Figure 3-5

1. **Level 0** : The Figure 3-5 considers that there are $|Nx| = 3$ total number of states (this parameter can be changed to any number). Given that $|Nx| = 3$ lets assume that the patient state at time t; X_t is given by

$$X_t = i;\ i \in \{1, 2, 3\}$$

2. **Level 1** : The Weibull distributions for each state $X_t = i;\ i \in \{1, 2, 3\}$ for ECG, ABP is shown right below the blue ECG, ABP observation circles in Figure 3-5. The $Y_{ECG},\ Z_{ABP}$ random variables are assumed to be generated from these state specific

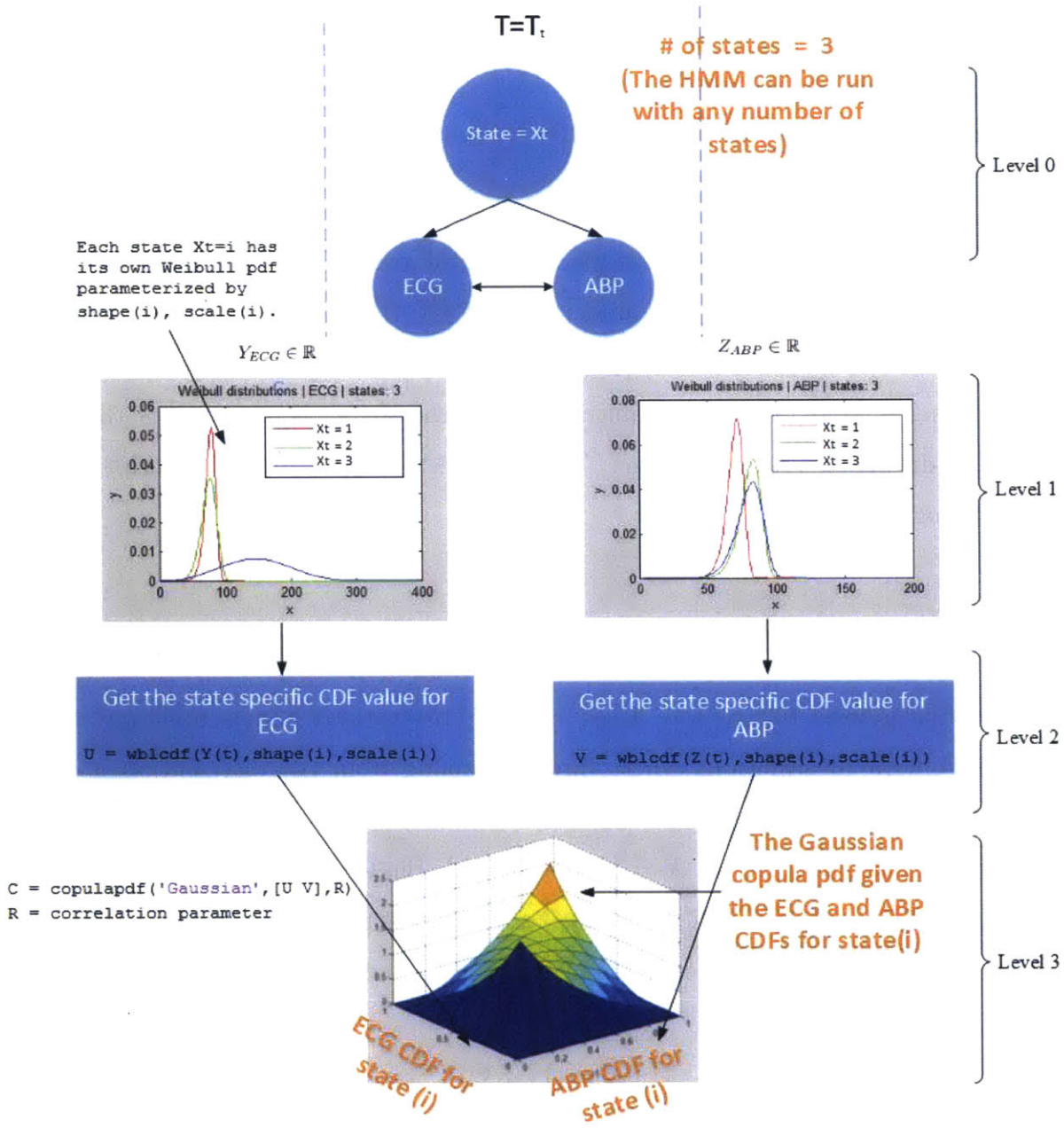


Figure 3-5: The view when zoomed into 1 time step

Weibull distributions. We chose to use the Weibull distribution to model the state specific distributions because Weibull is a very flexible pdf.

3. **Level 2** : Once we know the shape, scale values for a specific state $X_t = i$, we can construct the CDF of the Weibull distribution. Given the Weibull CDF and the training ECG, ABP data points we can get the associated quantiles that the training

points fall in. This is denoted as "Get state specific CDF value for ECG/ ABP" in Figure 3-5. The values that are output are in the range $[0, 1]$.

4. **Level 3** : We use the ECG, ABP CDF values and a Gaussian copula to model the joint behavior of ECG, ABP.

As mentioned earlier the steps described above are activated during each time step. A key intuition here is that even if the underlying ECG, ABP distributions are different for each patient we can use a Gaussian Copula to model their joint behavior.
Gibbs sampling is used as the mechanism to learn the shape(S), scale(λ) of the Weibull distributions during the training stage. The next section gives a brief description of the Gibbs sampling procedure.

3.5 Gibbs sampling

Gibbs sampling is used to generate a sequence of observations from a multivariate probability distribution in situations where direct sampling from the distribution is difficult [Bishop, 2006]. The distribution that we are interested in this thesis is the joint distribution between the hidden states $X = \{X_1, X_2, .., X_t\}$, the ECG, ABP observations (ECG): $Y = \{y_1, y_2, ..., y_t\}$, and (ABP): $Z = \{z_1, z_2, ..., z_t\}$ and the parameters Θ of the model. The parameters Θ consists of the scale(λ), shape(S) of ECG and ABP.

$$P(X, Y, Z, \Theta)$$

In a general sense, "Each step of the Gibbs sampling round will replace the value of one variable, drawn from the distribution of that variable conditioned on the values of the other variables" [Bishop, 2006]. After a variable is sampled it will be immediately used to update the distributions so that the next variable that's being sampled will see the result of the update. After all the variables are sampled and updated the next Gibbs sampling round would start. After a certain number of initial Gibbs sampling rounds (termed as "burn in" rounds) the sampled that are drawn can be used to approximate the joint and marginal distributions of interest.

Around 1000 Gibbs sampling rounds are performed while doing the experiments. The shape(s) and scale(λ) parameter values are seen to converge after around 20-30% of Gibbs sampling rounds are done. The convergence results are shown in Chapter 5. The setup for Gibbs sampling is shown in Figure 3-6.

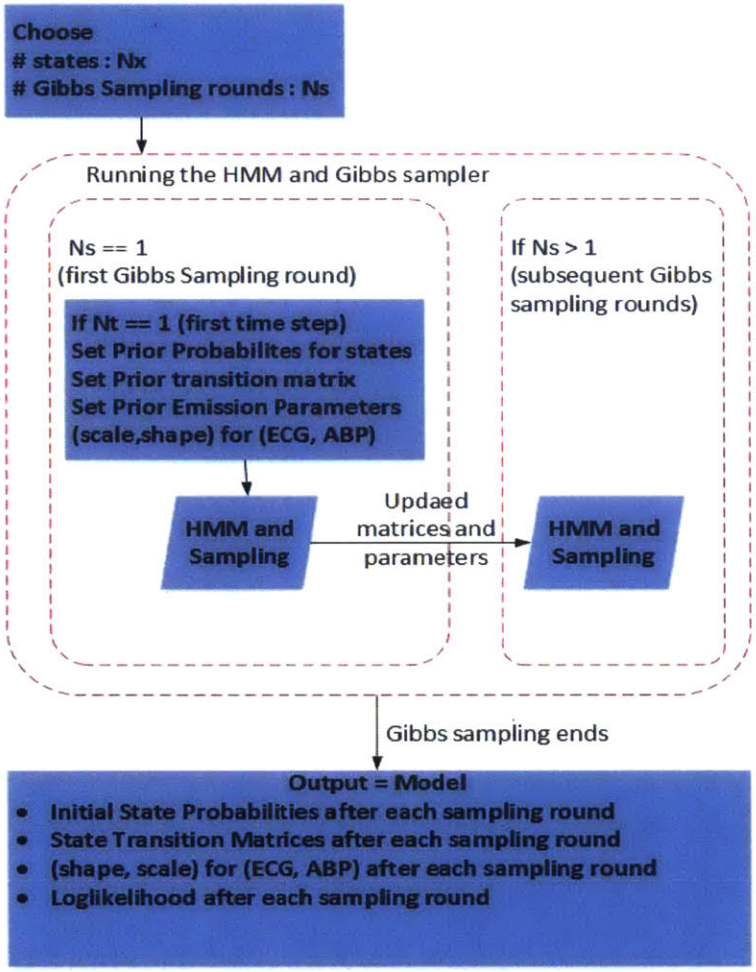


Figure 3-6: Gibbs sampling iterations.

At the beginning the number of states Nx and the number of Gibbs sampling rounds Ns is chosen by the user. For the very first time step in the first Gibbs sampling round the prior probabilities, transition matrix, emission probabilities are preset. For the subsequent steps the parameters are chosen by the Gibbs sampling process. The steps of the Gibbs sampling process are given below. The equations are from [Cuesta Infante and Veeramachaneni, 2015].

1. Sample from the initial state distribution. Our initial state distribution is chosen to

be a Dirichlet distribution.

$$p_D(\pi; 1 + I(x_1))$$

Where

- p_D - Dirichlet Distribution
- π - initial state probability distribution
- $I(x_1)$ - returns a row vector of $N_x - 1$ zeros and 1 one in the position of the active state

The equation is from [Cuesta Infante and Veeramachaneni, 2015].

2. Sample from each row of the transition matrix. Each row follows a Dirichlet distribution

$$p_D(\pi_{(i)}; 1 + K(x_t = i))$$

Where

- x_t - state at time t
- $a_{i,j}$ - transition matrix from state i to j
- $\pi_{(i)} = [a_{i,1}, \ldots, a_{i,N_x}]$ be the pmf over the transition from $x_t = i$ to every x_{t+1}.
- Function $K(x_t = i)$ returns a row vector where position j is the count of visits from $x_t = i$ to $x_{t+1} = j$, for $t = 1$ to T. Here $i, j \in \{1, 2, .., N_x\}$

The equation is from [Cuesta Infante and Veeramachaneni, 2015].

3. For each state(N_x) sample the relevant shape(S) and scale(λ) parameters for each variable from the relevant distribution. The equations below are for Y but can be generalized.

   For scale(λ):

   Assume the transformation $\lambda = \ell^{1/s}$

$$p_{IG}\left(\ell_i;\ \alpha + n_i, \beta + \sum_{t=1}^{T} \mathbf{I}(x_t = i) y_t^{s_i}\right), \tag{3.1}$$

Where

- x_t - state at time t
- p_{IG} - Inverse Gamma distribution
- α - parameter associated with the Inverse Gamma distribution
- β - parameter associated with the Inverse Gamma distribution
- n_i=$\sum_{t=1}^{T}\mathbf{I}(x_t=i)$ where $\mathbf{I}$ is the indicator function.
- y_t - observation
- s_i - scale related to $x_t = i$

The equation is from [Cuesta Infante and Veeramachaneni, 2015].

For shape(S):

$$p_G(s_i|Y_T) \propto f_E \cdot p_G(s_i, \zeta', \xi'); \tag{3.2}$$

Where

- p_G - Gamma distribution
- ζ' - parameter associated with the Gamma distribution
- ξ' - parameter associated with the Gamma distribution
- s_i - scale related to $x_t = i$
- y_t - observation
- ℓ - taken from the transformation $\lambda = \ell^{1/s}$
-

$$f_E = \exp\left(-\frac{1}{\ell_i}\sum_{t=1}^{T}\mathbf{I}(x_t = i)y_t^{s_i}\right). \tag{3.3}$$

The equations are from [Cuesta Infante and Veeramachaneni, 2015].

4. Sample from the Markov chain of states

$$p(x_1 = i|\cdots) \propto \pi_i \cdot p_O(y_1, z_1|x_1 = i) \cdot p(y_{2:N_t}, z_{2:N_t}|x_1 = i) \tag{3.4}$$

$$p(x_t = j|x_{t-1}=i, \cdots) \propto a_{ij} \cdot p_O(y_t, z_t|x_t=j) \cdot p(y_{t+1:N_t}, z_{t+1:N_t}|x_t = j) \tag{3.5}$$

Where

- $p_O(y_t, z_t|x_t = i)$ is the observation probability of the pair (y, z) at time t, given that state $x_t = i$.
- $\pi_{(i)} = [a_{i,1}, \ldots, a_{i,N_x}]$ be the pmf over the transition from $x_t = i$ to every x_{t+1}.
- $a_{i,j}$ - transition matrix from state i to j

The equations are from [Cuesta Infante and Veeramachaneni, 2015].

5. Store the sampled variables.

6. Go back to step 1.

The Gibbs sampling section concludes the description of the Latent State Space Copula Model. The prediction framework will be introduced in the next section where we use the Latent State Space Copula Model to make predictions on whether the patient blood pressure levels are going to be below a certain threshold in a future time window.

3.6 Model Prediction

The blood pressure prediction problem is shown in Figure 3-7.

If the current time is t, the historical signals of γ time steps before is used to train the machine learning algorithm. The prediction is made for τ period of time steps ahead, for a

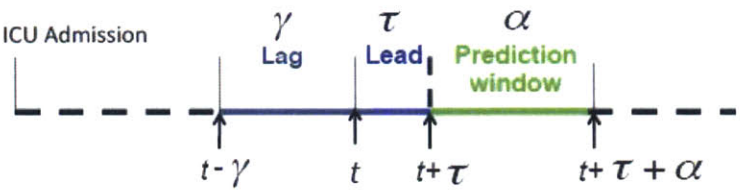


Figure 3-7: The blood pressure prediction problem. Figure taken from [Waldin, 2013]

time window of length α. This is the fundamental prediction problem and over the past few years there have been attempts to predict blood pressure using various algorithms based on this prediction problem definition [Waldin, 2013], [Dernoncourt et al.,], [Kim et al., 2014].

For the Latent State Space Copula Model the implementation of the prediction process is given below.

We run a large number(i.e $\sim$ 1000) of Gibbs sampling rounds and store the updated models. The following parameters are extracted from the final Gibbs sampling round

1. shape(λ), scale(s) parameters of the Weibull pdfs associated with ECG, ABP
2. transition matrix(A), initial matrix ($Initial$) of the Hidden Markov Model

With these parameters we make predictions on a test data set. We train on 500 patients and test on 150 patients. Our prediction problem is to observe lag number of historical times steps and predict $lead$ time steps ahead for a window of w length. The current results are for

1. $lag = 6 \rightarrow 120min$
2. $lead = 1 \rightarrow 20min$
3. window length $w = 1 \rightarrow 20min$

In the text below $Y = ECG,\ Z = ABP,\ X = states\{1, 2, .., N_x\}$. Consider the time to go form $t = 1, 2, ...n$. The algorithm for making predictions is given below.

At time step k;

- Step 1 :

  Compute $P(X_k|Y_1^{k-1}, Z_1^{k-1})$ using the parameters learnt from the HMM training phase;

$P(X_k|Y_1^{k-1}, Z_1^{k-1})$ is a N_x x 1 vector (i.e If $N_x = 3$ then $P(X_k|Y_1^{k-1}, Z_1^{k-1})$ is a 3 x 1 vector).

- Step 2 :

  Use $P(X_k|Y_1^{k-1}, Z_1^{k-1})$ and the $tranistionmatrix(A)$ from the final Gibbs sampling round to predict a $lead + window$ time ahead;

  $P(X_{k+lead+window}|Y_1^{k-1}, Z_1^{k-1}) = P(X_k|Y_1^{k-1}, Z_1^{k-1}) * A^{lead+window}$

  We assume that the transition matrix to be the same for the period of $lead + window$ number of time steps. Additionally since the window size = 1, we can do a simple addition in the exponent of $A^{lead+window}$ when projecting ahead. If the $windowsize > 1$ the process would be different.

- Step 3 :

  Compute the $P(Z_{k+lead+window} < 60_{mmHg}|Y_1^{k-1}, Z_1^{k-1})$;

  In order to compute this value we need to use the precomputed values of $P(Z < 60_{mmHg}|X; \lambda_Y, S_Y, \lambda_Z, S_Z)$. The steps to pre compute this value is given below

  - Given the $\lambda_Y, S_Y, \lambda_Z, S_Z$ parameters taken from the final Gibbs sampling round we first find $P(Y, Z|X) = C(U, V; Q) * P(Y|X) * P(Z|X)$, where C is a Gaussian Copula, $U = CDF_{marginal}(Y), V = CDF_{marginal}(Z)$ and Q a covariance matrix for a given state X
  - Then we marginalize to get

    $$P(Z|X) = \int_{y\_min}^{y\_max} P(Y, Z|X)dy$$

  - We can get the probability of Z being less than 60 mmHg by

    $$P(Z < 60_{mmHg}|X) = \int_0^{60} P(Z|X)dz$$

    Since the scale(λ),shape(s) are parameters are known immediately after training this value can be precomputed. The precomputed marginals and cdf's are shown in figure 3-8 and 3-9

– This value is the same for any time step. Thus

$$P(Z_{k+lead+window} < 60_{mmHg} | X_{k+lead+window}) = P(Z < 60_{mmHg} | X)$$

Finally we use the pre-computed $P(Z_{k+lead+window} < 60_{mmHg} | X_{k+lead+window})$ value to calculate

$$P(Z_{k+lead+w} < 60_{mmHg} | Y_1^{k-1}, Z_1^{k-1}) = \sum_{X_{k+lead+w}=1}^{N_x} P(Z_{k+lead+w} < 60_{mmHg} | X_{k+lead+w}) * P(X_{k+lead+w} | Y_1^{k-1}, Z_1^{k-1}) \quad (3.6)$$

This is our chosen method of implementing prediction

- Step 3 (Alternative)

  Another option for step 3 is

  Given the observations find the most likely state and associated probability from step 2

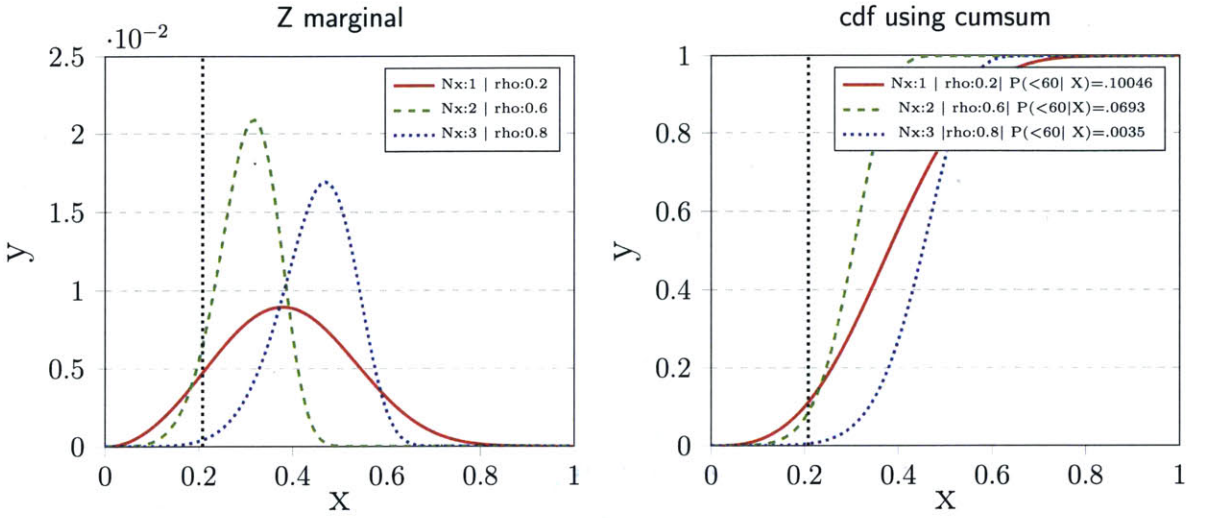
  $P(\hat{X} | Y_1^{k-1}, Z_1^{k-1}) = argmax_X P(X_{k+lead+window} | Y_1^{k-1}, Z_1^{k-1})$

  Given $P(\hat{X} | Y_1^{k-1}, Z_1^{k-1})$ and the pre-computed $P(Z < 60_{mmHg} | X)$ find the joint probability of $\hat{X}$ and $ABP < 60mmHg$ conditioned on the observations

  $P(Z_{k+lead+w} < 60_{mmHg}, \hat{X} | Y_1^{k-1}, Z_1^{k-1}) = P(Z_{k+lead+w} < 60_{mmHg} | \hat{X}) * P(\hat{X} | Y_1^{k-1}, Z_1^{k-1})$

The marginal distributions and cdf's for step 3 is given below.

1. Marginal distributions and cdf's $N_x = 3$

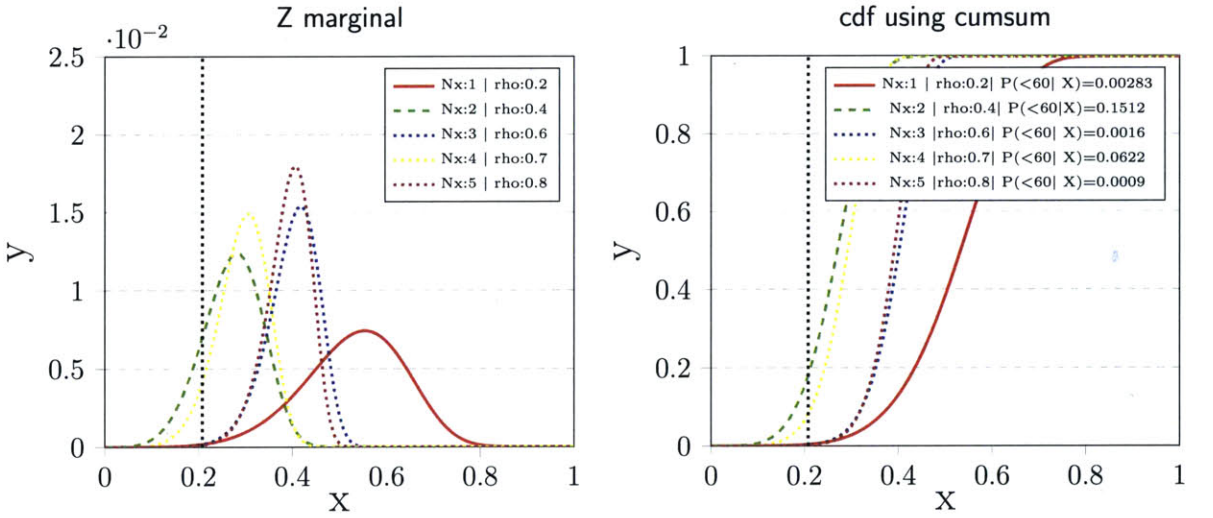


(a) The precomputed marginal distributions based on the joint. The dashed black line is the ABP= 60mmHg.

(b) The precomputed cdf's based on the marginals.The dashed black line is the ABP=60mmHg.

Figure 3-8: Marginal distributions and CDFs based on the marginals

2. Marginal distributions and cdf's $N_x = 5$



(a) The precomputed marginal distributions based based on the joint. The dashed black line is the ABP= 60mmHg.

(b) The precomputed cdf's based on the marginals.The dashed black line is the ABP=60mmHg.

Figure 3-9: Marginal distributions and CDFs based on the marginals

The code for pre-computing $P(Z < 60_{mmHg}|X; \lambda_Y, S_Y, \lambda_Z, S_Z)$ is given in Appendix A.

Basically the key takeaway for this section is that the equation

$$P(Z_{k+lead+w} < 60_{mmHg}|Y_1^{k-1}, Z_1^{k-1}) = \sum_{X_{k+lead+w}=1}^{N_x} P(Z_{k+lead+w} < 60_{mmHg}|X_{k+lead+w}) * P(X_{k+lead+w}|Y_1^{k-1}, Z_1^{k-1}) \quad (3.7)$$

which allows us to make predictions using the Latent State Space Copula Model. This model was implemented in MATLAB and tested on the patient data set mentioned in Chapter 2.

This concludes our discussion of Model Description and Prediction and we will now move on to explain the experiments in Chapter 4 and the results analysis in Chapter 5.

Chapter 4

Experiments

This chapter discusses the experiment setup for running the Latent State Space Copula Model and Classical Discrete HMM on the patient data set. Other various methodologies that we utilized while doing the experiments will also be discussed. The chapter will end with a brief mention of the computing costs associated with the experiments.

4.1 Data splits and 5 fold crossvalidation

We split our 650 patient data set into 150 patients (23.1%) as the withheld data set and perform 5 fold cross validation on the rest of the 500 patients (76.9%). 5 fold crossvalidation gives us an idea about the variance within the training data set and also an idea about how well our models are going to generalize. The experiments were run with number of states $N_x = \{3, 5\}$. The following procedure was used when running the experiments

1. Step 1: Run 1000 sampling rounds on training folds and save the model.

2. Step 2: From the saved model get the final $shape(s)$, $scale(\lambda)$ from the last Gibbs sampling iteration.

3. Step 3: Run the prediction algorithm on the testing folds using the $shape(s)$, $scale(\lambda)$ from step 2

In addition to cross validation there was training on the complete training data (500 patients) and the prediction algorithm was run on the withheld data (150 patients). The ROC curves for each state and the analysis is given in Chapter 5.

To compare the Latent State Space Copula Model with the Discrete HMM we ran the discrete HMM for $N_x = \{3, 5\}$ on the same data set. The following procedure was used to generate the plots for the Discrete HMM

1. Select the number of states Nx=$\{3, 5\}$
2. Select lead = 20 minutes
3. Run the Discrete HMM for various lags = $20, 40, 60, 80, 100, 120$ min

The results and analysis for each state for various lags is given in Chapter 5.

4.2 Parallel Data processing

When running experiments with the Latent State Space Copula Model , (ECG, ABP) readings from 500 patients needed to be processed for each time step. Considering the fact that there are approximately 500 time steps for each patient, this computation would take a long time if done sequentially. In order to speed up the computation and to make use of multicore computer clusters (MIT CSAIL provides compute nodes with 16-24 cores) we utilized the "parallel for loops" programming paradigm. This essentially involves parallel processing of "n"number of patients at a time. Data from a single patient would be assigned a single core in a multicore computer cluster environment. By using this paradigm we were able to bring down the processing time by a factor of $1/num\_cores$.

The basic structure of a parfor loop is given below. MATLAB and python offer parallel for loop capabilities and for this thesis the "parfor" loops from the MATLAB programming environment were used.

The MATLAB parfor variable classifications are shown below

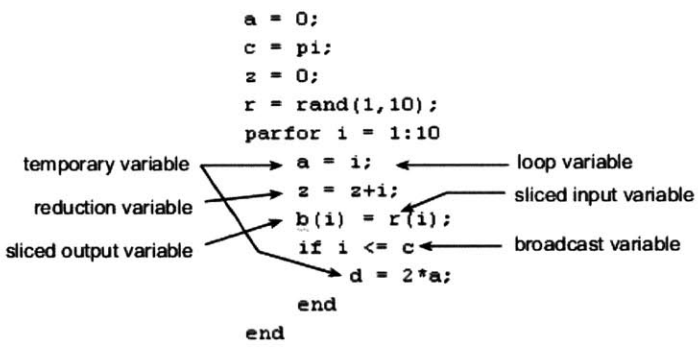


Figure 4-1: MATLAB parfor variable classification. Figure taken from [The MathWorks, 2015].

The definition of each type of variable and how it was used in the context of medical data is given below. Source [The MathWorks, 2015].

1. Loop Variable : This is the variable that's considered as the index for each parallel run of the parfor loop. In our context each patient p was used as the loop index variable for each parallel run.

2. Broadcast Variables: These variable values are broadcast to all parallel workers. The value is assigned before the loop starts and is used without modification inside the loop. The model (Θ) varibales were used as broadcast variables during the prediction phase.

3. Sliced variables : Arrays or data structures that store values generated or used when the loop runs. The ROC probability values were stored in a sliced variable array during the prediction phase.

4. Reduction Variables : Variables that accrue values as the loop runs, disregarding execution order. A reduction variable was used to count the number of patients that were currently processed.

5. Temporary Variables : Variables created and used inside the loop.

4.3 Computation cost

During the training stage of the HMM 1000 Gibbs sampling rounds were run for states $N_x = \{3, 5\}$. For $N_x = 3$ a single Gibbs iteration took 10 minutes on average on a cloud node that has 24 cores, 24 GB RAM when running for 400-500 patients in parallel. Thus 1000 Gibbs sampling rounds takes 10*60*1000/(24*3600) = 6.94 days. From another view point each Gibbs sampling round deals with 400-500 patients. Thus it took 10 minutes to process 400-500 patients for $N_x = 3$ during 1 Gibbs sampling round. This makes the time to process a single patient to be roughly 10*60/500= 1.2 seconds during 1 Gibbs iteration. For $N_x = 5$ a single Gibbs iteration took 15 minutes on average on a cloud node that has 24 cores, 24 GB RAM when running for 400-500 patients in parallel. Thus 1000 Gibbs sampling rounds takes = 15*60*1000/(24*3600) = 10.41 days and time to process a single patient within 1 Gibbs iteration is = 15*60/500 = 1.8 seconds.

During testing the models generated from training were extracted an evaluated on the testing data set. During the testing stage we were able to use the MATLAB parfor paradigm and cut the processing down to $1/num\_cores = 1/24$ *cores*.

After painting a clear picture of the experiments we now move on to Chapter 5 where we discuss the results from the experiments.

Chapter 5

Results and Discussion

Our novel Latent State Space Copula Model and the associated Gibbs sampling process provides a set of parameters associated with the Weibull distributions at the end of a complete Gibbs sampling simulation. In this chapter we discuss how we select the parameter values that accurately model the underlying (ECG, ABP) distributions, evaluate how well they fit the data using the loglikelihood function, see how well they perform when doing Acute Hypotensive Episode (AHE) prediction and compare them to a Classical Discrete HMM which was introduced in Chapter 3. This chapter addresses the following high level questions

1. How can we evaluate the convergence characteristics of the models generated by the Latent State Space Copula Model? (A discussion and definition of convergence will be given in section 5.1)

2. How good are the models in terms of predicting Acute Hypotensive Episodes? (Addressed in Section 5.2)

3. How does the model compare to the Classical Discrete HMM? (Addressed in Section 5.3)

5.1 Convergence

During the Gibbs sampling process we sample parameters shape(S), scale(λ) from prior and posterior distributions. If we are sampling k parameters , each parameter is visited "in or-

der" within a single Gibbs sampling round.

In this context, convergence means when the number of sampling rounds N_s is made very large (the exact number is problem specific) the samples we draw would be very close to samples drawn from the actual marginal and joint distributions [Gelfand and Smith, 1990]. This is rigorously proven in the original Gibbs sampling paper [Geman and Geman, 1984] and summarized in [Gelfand and Smith, 1990]. Another very intuitive way of interpreting convergence is to consider the Gibbs sampling process as a Markovian update scheme where a Markov chain of samples is generated (each sample is correlated with the nearby sample). The Markov chain achieving steady state after a large number of state transitions is analogous to the Gibbs sampler producing samples that approximate the joint distribution after a large number of Gibbs sampling rounds [Casella and George, 1992].

In this thesis we define the parameters to have converged when the shape(S), scale(λ) values of the ECG, ABP Weibull distributions show approximately the same value for a large number of iterations with minimum fluctuation. We can observe this by looking at convergence statistics for each state for a given experiment.

Convergence statistics for each state

The simplest way to observe this is by observing the parameter values achieve a steady value as seen in Figure 5-1 and Figure 5-2 for $Nx = 3$ and in Figure 5-8 and Figure 5-9 for $N_x = 5$. But to be more precise we constructed a series of Tables [5.1 - 5.8] to quantify convergence statistics. Since there are parameters associated with multiple states and some state associated parameters achieve convergence faster than others we decided to look at parameter values for each state separately and evaluate convergence. We then try to come up with a common threshold for each group of parameters. Namely $\{T\_ecg\_scale,\ T\_ecg\_shape,\ T\_abp\_scale,\ T\_abp\_shape\}$. If the standard deviation of the associated parameter value is equal to or below this threshold for a I number of iterations then we consider the parameter values to have converged. We observe the tables and propose values for the suggested parameters.

In the tables the first column is the iteration number. The next columns are the mean and standard deviations for the scale (S) and shape (λ) values for ECG, ABP.

1. $N_x = 3 \mid state = 1$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.423	0.028	2.981	0.352	0.397	0.031	3.343	0.52
201-400	0.432	0.002	3.047	0.035	0.432	0.002	2.987	0.013
401-600	0.428	0.001	2.975	0.016	0.435	0.001	2.99	0.014
601-800	0.428	0.001	2.977	0.021	0.435	0.001	2.983	0.011

Table 5.1: The Mean and standard deviation of the $shape(S), scale(\lambda)$ for $N_x = 3 \mid state = 1$

2. $N_x = 3 \mid state = 2$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.214	0.013	5.421	0.991	0.372	0.033	5.509	0.805
201-400	0.221	0	6.183	0.032	0.331	0.002	5.369	0.1
401-600	0.22	0	6.173	0.012	0.329	0.001	5.532	0.038
601-800	0.22	0	6.138	0.022	0.33	0.001	5.544	0.036

Table 5.2: The Mean and standard deviation of the $shape(S), scale(\lambda)$ for $N_x = 3 \mid state = 2$

3. $N_x = 3 \mid state = 3$

If we observe the standard deviation for shape (S) for (ECG,ABP) we see that after the 400th Gibbs iteration the standard deviation for the shape is below 0.061 for the next

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Itera-tion	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.228	0.043	4.111	1.269	0.563	0.143	6.011	0.578
201-400	0.206	0	5.705	0.044	0.487	0.004	6.045	0.164
401-600	0.206	0	5.725	0.03	0.482	0.002	6.421	0.061
601-800	0.206	0	5.715	0.029	0.483	0.002	6.55	0.058

Table 5.3: The Mean and standard deviation of the $shape(S), scale(\lambda)$ for $N_x = 3 \mid state = 3$

400 iterations and the mean is approximately constant for each state. We also see that the standard deviation for scale (λ) for (ECG,ABP) is below 0.002 after the 400th iteration for a 400 iteration period while the mean remains approximately constant. These observations motivate us to offer a simple definition of convergence in the following paragraphs.

To observe this pattern further, we look at Tables [5.4 - 5.8] where the shape(S), scale(λ) values for (ECG,ABP) related to each state for $Nx = 5$ is shown.

1. $N_x = 5 \mid state = 1$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Itera-tion	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.423	0.028	2.981	0.352	0.397	0.031	3.343	0.52
201-400	0.432	0.002	3.047	0.035	0.432	0.002	2.987	0.013
401-600	0.428	0.001	2.975	0.016	0.435	0.001	2.99	0.014
601-800	0.428	0.001	2.977	0.021	0.435	0.001	2.983	0.011

Table 5.4: The Mean and standard deviation of the $shape(S), scale(\lambda)$ for $N_x = 5 \mid state = 1$

2. $N_x = 5 \mid state = 2$

3. $N_x = 5 \mid state = 3$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.214	0.013	5.421	0.991	0.372	0.033	5.509	0.805
201-400	0.221	0	6.183	0.032	0.331	0.002	5.369	0.1
401-600	0.22	0	6.173	0.012	0.329	0.001	5.532	0.038
601-800	0.22	0	6.138	0.022	0.33	0.001	5.544	0.036

Table 5.5: The Mean and standard deviation of the $shape(S)$, $scale(\lambda)$ for $N_x = 5 \mid state = 2$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.228	0.043	4.111	1.269	0.563	0.143	6.011	0.578
201-400	0.206	0	5.705	0.044	0.487	0.004	6.045	0.164
401-600	0.206	0	5.725	0.03	0.482	0.002	6.421	0.061
601-800	0.206	0	5.715	0.029	0.483	0.002	6.55	0.058

Table 5.6: The Mean and standard deviation of the $shape(S)$, $scale(\lambda)$ for $N_x = 5 \mid state = 3$

4. $N_x = 5 \mid state = 4$

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.214	0.013	5.421	0.991	0.372	0.033	5.509	0.805
201-400	0.221	0	6.183	0.032	0.331	0.002	5.369	0.1
401-600	0.22	0	6.173	0.012	0.329	0.001	5.532	0.038
601-800	0.22	0	6.138	0.022	0.33	0.001	5.544	0.036

Table 5.7: The Mean and standard deviation of the $shape(S)$, $scale(\lambda)$ for $N_x = 5 \mid state = 4$

5. $N_x = 5 \mid state = 5$

If we observe the standard deviation for the shape(S) for (ECG,ABP) we see that after

	ECG ($Nx = 3$)				ABP ($Nx = 3$)			
	Scale		Shape		Scale		Shape	
Iteration	mean	std.dev	mean	std.dev	mean	std.dev	mean	std.dev
1-200	0.228	0.043	4.111	1.269	0.563	0.143	6.011	0.578
201-400	0.206	0	5.705	0.044	0.487	0.004	6.045	0.164
401-600	0.206	0	5.725	0.03	0.482	0.002	6.421	0.061
601-800	0.206	0	5.715	0.029	0.483	0.002	6.55	0.058

Table 5.8: The Mean and standard deviation of the $shape(S), scale(\lambda)$ for $N_x = 5 \mid state = 5$

the 400th Gibbs iteration the standard deviation remains below 0.061 and the mean is approximately constant. We also see that the standard deviation for the scale(λ) is below or approximately equal to 0.002 after the 400th iteration while the mean remains approximately constant in that period. We also see that 400 iterations is a sufficiently large enough period to observe to detect parameter convergence.

Based on these observations if the mean value of the parameters for each state was constant for a $I = 400$ iteration period and the standard deviation of parameters for each state is less than or equal to a certain threshold I consider the parameter values to have converged. The threshold values for each set of parameters is given below

$T\_ecg\_scale = .002$

$T\_abp\_scale = .002$

$T\_ecg\_shape = .061$

$T\_abp\_shape = .061$

$I = 400\ iterations$

Another alternative way to define convergence would be to use statistics averaged over all states instead of looking at each state specifically.

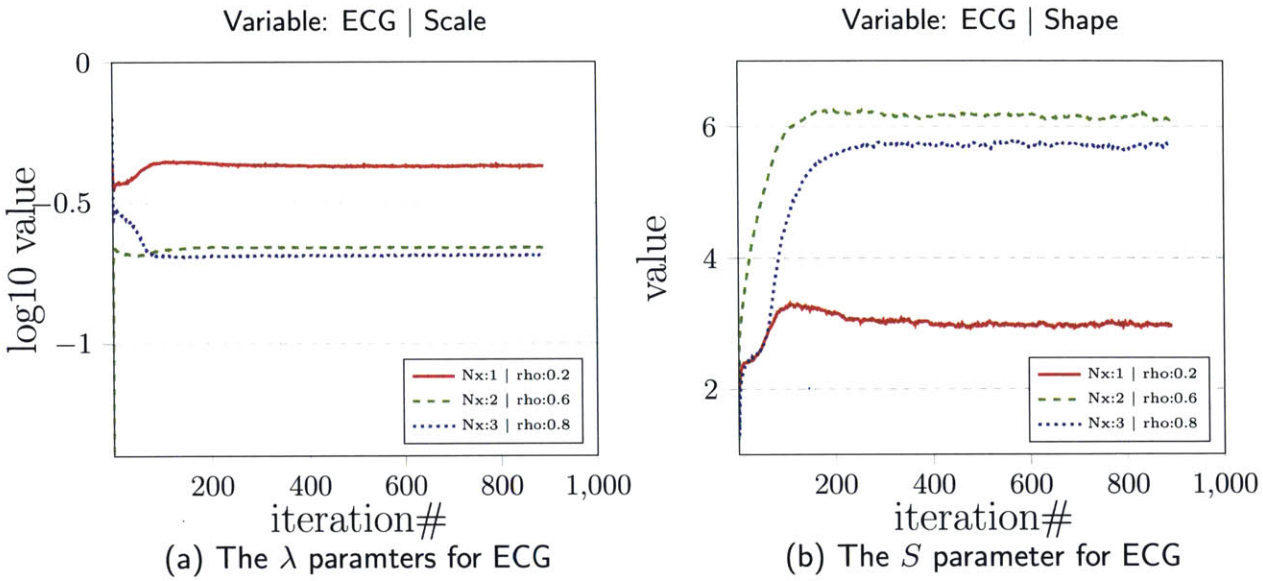
After convergence there are various ways to pick the final parameter values. In this thesis after observing that the parameter values have converged we extract the parameter values

from the last Gibbs sampling iteration and use it to recreate the underlying distributions.

Another good approach to discuss convergence is to look at cases where there is "non-convergence". In general we would observe the parameter values to show large fluctuations or oscillate when the Gibbs sampler is in a "non-convergent" state. This might occur in the initial rounds of a Gibbs sampling process (i.e the first 5% of the iteration in Figure 5-1, Figure 5-2). Additionally "non-convergence" might also occur when the Gibbs sampler is exploring multiple locally optimum values and moving through them before reaching the global optimum(if it reaches the global optimum at all).
The convergence results for $N_x = \{3, 5\}$ and a discussion is given below.

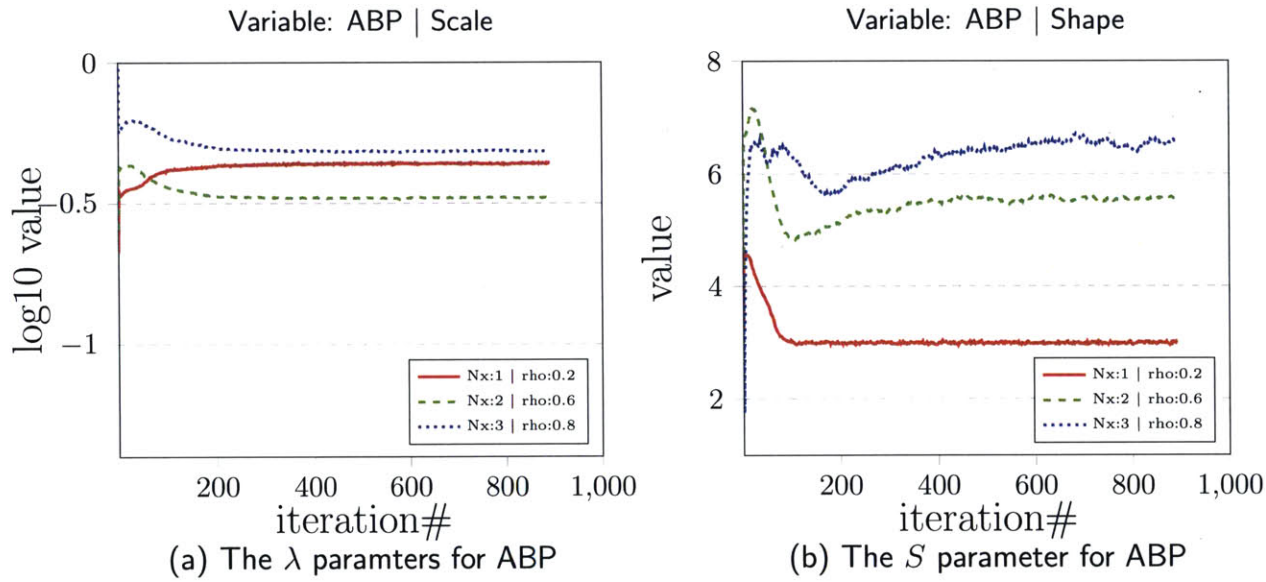
Results for $N_x = 3$

The values of the $scale(\lambda), shape(S)$ Weibull parameters for ECG and ABP are seen to converge after about 400-600 Gibbs iterations based on Table [5.1 - 5.3] and the definition provided in section 5.1. This can also be visually observed in Figure 5-1 and Figure 5-2. Notice that generally the shape(S) values for both ECG, ABP is greater than 1 which leads to a Weibull plot that has the shape of a Gaussian or Rayleigh distribution.



(a) The λ paramters for ECG (b) The S parameter for ECG

Figure 5-1: ECG scale and shape as iterations progressed.



(a) The λ paramters for ABP (b) The S parameter for ABP

Figure 5-2: ABP scale and shape as iterations progressed.

The convergence of $scale(\lambda), shape(S)$ parameters can be further verified by looking at the standard deviation of the values over iterations in Tables [5.1 - 5.3]. We see the standard deviations decrease and stabilize as the number of iterations increase.

As mentioned above the shape and scale value from the last Gibbs sampling round is

extracted and used to plot the graphs in Figure 5-3. In Figure 5-3 the following scale is used

- ECG : $[0, 1] \rightarrow [\ .0082\ ,\ 359.63\ ]$ bpm
- ABP : $[0, 1] \rightarrow [\ 31.81\ ,\ 167.39\ ]$ mmHg
- ABP : 60 $mmHg\ = 0.2079$

Figures 5-3 (a) and (b) shows the state specific Weibull pdfs. When the state specific pdfs are plotted together the resultant distribution approximately matches the original ECG, ABP distributions. This can be observed from Figure 5-4 and Figure 5-5 where the overlapped Weibull plots are placed side by side with the original ECG, ABP plots. The fact that we see the parameter values converging in Figures 5-1, 5-2 and the fact that we see the original distributions approximately recreated in Figure 5-4, 5-5 provides us useful feedback about the overall convergence of our Gibbs sampler.

The figure in 5-3 (c) is the cdf of the univariate ABP distributions. The black vertical line in 5-3 (c) is the $ABP = 60mmHg$ (associated with $x = 0.2079$). By observing Figure 5-3 (c) we can state that a patient having his/her $ABP < 60mmHg$ for a 20 minute interval, has a higher probability of being in state 1 as opposed to state 2 or 3.

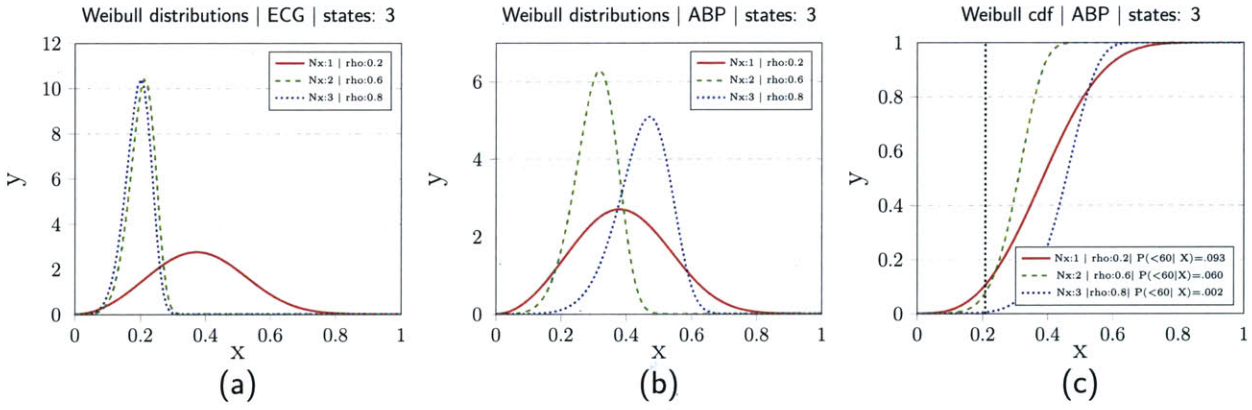


Figure 5-3: #Patients: 500 | #Sampling iterations: 892 | ECG,ABP pdfs, cdf

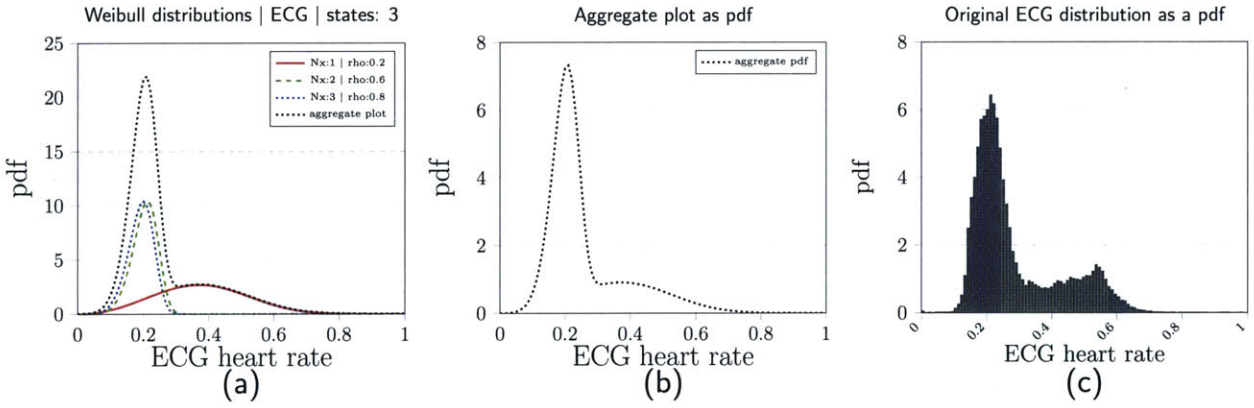


Figure 5-4: Nx = 3 | Comparison of aggregate pdf vs original pdf for ECG

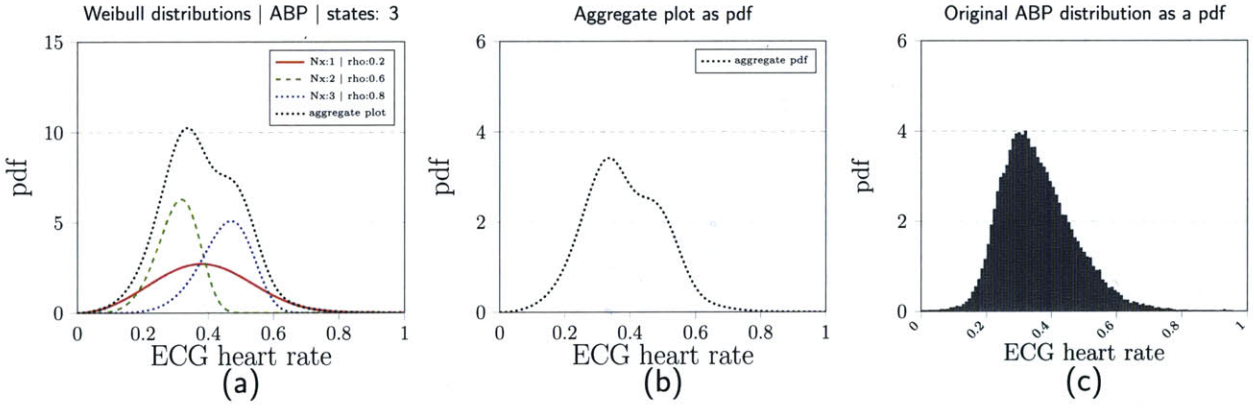


Figure 5-5: Nx = 3 | Comparison of aggregate pdf vs original pdf for ABP

The figures above gave us an idea of the convergence of parameters. Next we look into how the latent states are distributed after the parameters have converged. The distribution of states in the time series of the last Gibbs sampling round is shown in figure 5-6.

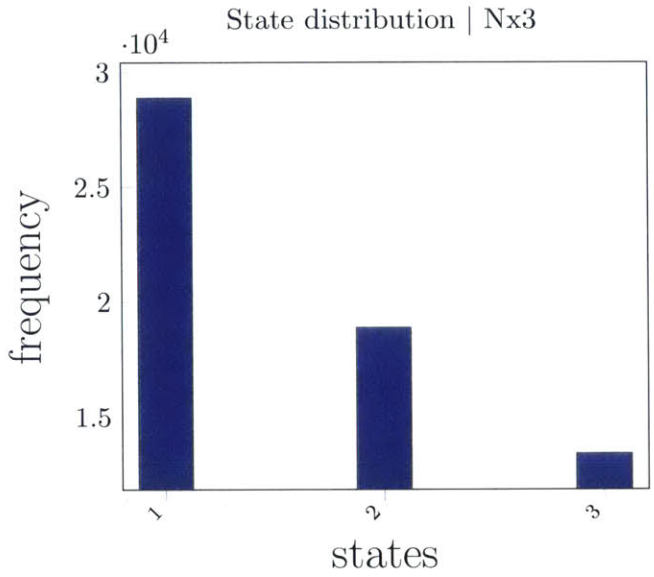


Figure 5-6: State distribution over time slices for all patients| patients = 500 | sampling iterations = 892 | total # states = 61135

There are a total number of 61135 state associated with 61135 time slices. Given that there are 500 patients there are an average number of $61135/500 = 122.27$ states or time slices per patient. Given that each state or time slice is associated with a 20 minute interval this is equivalent to $122.27 * 20/(60 * 24) = 1.6982$ days worth of readings. We see that the majority of the patients are in state 1. By observing Figure 5-3 we can see that state 1 is associated with a high ECG and medium ABP values (based on the modes of the ECG, ABP distributions for state 1: in red).

Table 5.9 shows the statistics related to the $scale(\lambda), shape(s)$ values over all the Gibbs sampling rounds. This table was populated by recording the $scale(\lambda), shape(s)$ value for each ECG, ABP state at the end of each Gibbs iteration and calculating the statistics (standard deviation and mean) for each parameter. We observe that the scale(λ) has lower standard deviations than the shape. Our choice of using the Weibull distribution gives us more flexibility in exploring different shapes of distributions (exponential, Gaussian, Raleigh).

ECG ($Nx = 3$)				
	Scale		Shape	
State	mean	std.dev	mean	std.dev
1	0.428	0.014	2.993	0.17
2	0.219	0.007	5.996	0.562
3	0.211	0.022	5.355	0.899
ABP ($Nx = 3$)				
	Scale		Shape	
State	mean	std.dev	mean	std.dev
1	0.426	0.022	3.067	0.287
2	0.339	0.024	5.495	0.391
3	0.502	0.075	6.284	0.371

Table 5.9: Mean and standard deviation over all iterations for estimated parameters ($Nx = 3, Ns = 892$)

Finally we evaluate the loglikeliood fit of our model to the data. Suppose there are t observations $y_1, y_2, ..., y_t$ which come from an unknown probability distribution that is parametrized by Θ (vector of parameters) the likelihood L is defined as

$$L(\Theta \ ; \ y_1, y_2, ..., y_t) \ = \ P(y_1, y_2, ..., y_t \mid \Theta)$$

In order to get the log likelihood we take the logarithm of the above function. In our model we do not assume that the data is independent. Thus based on our structure introduced in Chapter 3 we use a copula based approach to calculate the loglikelihood. The likelihood for our model is

$$L(\Theta; y_1, y_2, ..., y_t) = P(y_1|\Theta) * P(y_2|\Theta) * ... * P(y_t|\Theta) * copula\_pdf(y_1, y_2, ..., y_t)$$

$$\ln L(\Theta; y_1, y_2, ..., y_t) = \sum_{i=1}^{t} lnP(y_i|\Theta) + \ln copula\_pdf(y_1, y_2, ..., y_t)$$

In our example there are 2 variables $Y = ECG$, $Z = ABP$ and each follow a Weibull distribution with shape S and scale λ. For simplicity we assume that we are computing the equations for a specific state i.

W - denotes the cumulative distribution function of the Weibull pdf

c - denotes the Gaussian copula pdf

ρ - denotes a correlation parameter associated with the copula

$U = W(Y; S_y, \lambda_y)$

$V = W(Z; S_z, \lambda_z)$

$\rho = [1\ \rho_i;\ \rho_i\ 1]$

the copula_pdf can be wriiten as

$$copula\_pdf(Y, Z) = c\ (U, V; \rho)$$

The individual marginal pdfs $P(y_i|\Theta)$ can be written as

$$P\ (Y|\Theta) = Weibull(Y|S_y, \lambda_y)$$

$$P\ (Z|\Theta) = Weibull(Z|S_z, \lambda_z)$$

The loglikelihood equation can be rewritten as

$$\ln L(\Theta; y_1, y_2, ..., y_t) = \ln Weibull(Y|S_y, \lambda_y) + \ln Weibull(Z|S_z, \lambda_z) + \ln c\ (U, V; \rho)$$

With the above equations we compute the loglikelihood fit of our model to the data. The resulting plot is given in Figure 5-7. The likelihood is a measure of proportionality of the probability of observing the data given the model. Thus it is a relative measure and the actual value of a point measure of the loglikelihood does not carry any interpretable value. It can also have positive values [Eliason, 1993]. We observe that as the sampling iterations increase the loglikelihood improves suggesting that the relative probability of observing the

data given the model improves.

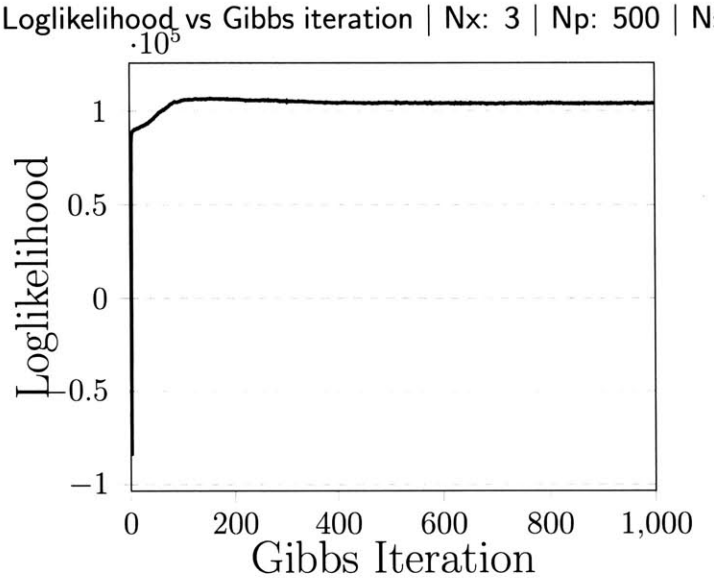
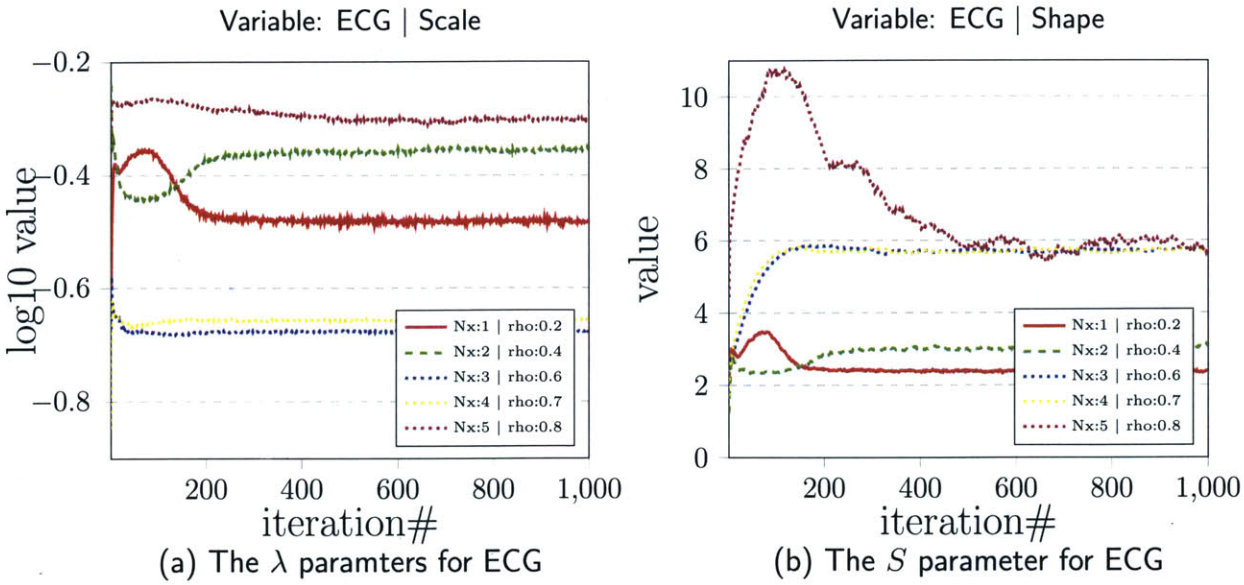


Figure 5-7: Loglikelihood vs Gibbs iteration for Nx: 3

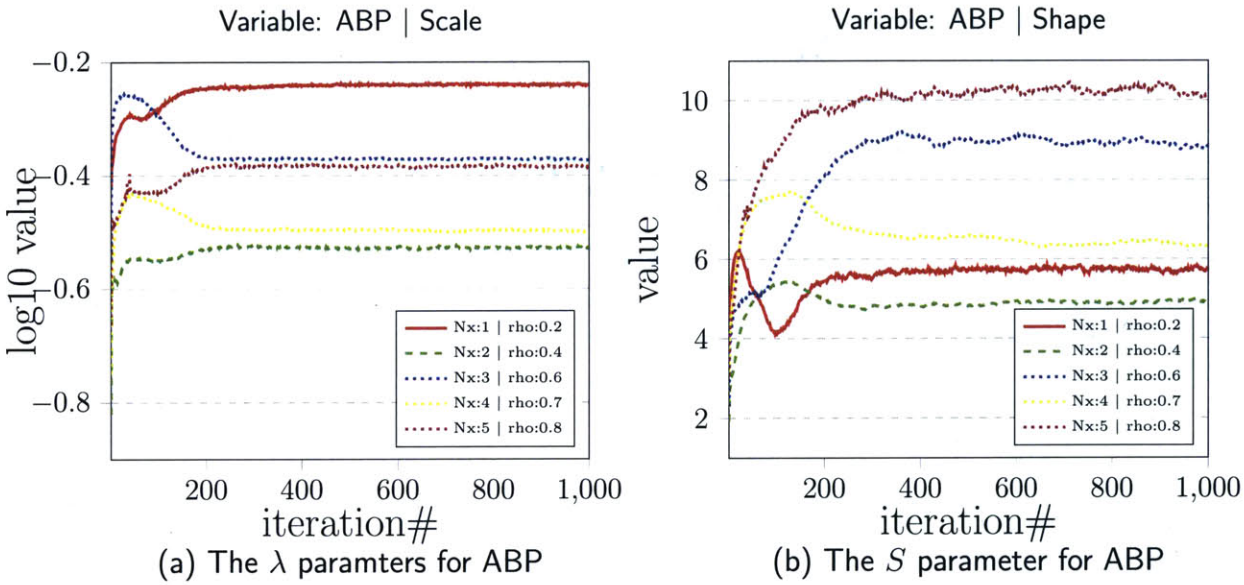
Results for $N_x = 5$

From Tables [5.4 - 5.8] we see that for $N_x = 5$ it takes around 400-600 Gibbs iterations for the parameter values to converge (based on the definition of convergence given in section 5.1). This is approximately similar to the number of iterations needed for $N_x = 3$ to converge. The convergence behavior can also be visually observed from Figure 5-8 and Figure 5-9 where we see the parameter values converge around the 400-600 iteration range.



(a) The λ paramters for ECG (b) The S parameter for ECG

Figure 5-8: ECG scale and shape as iterations progressed.



(a) The λ paramters for ABP (b) The S parameter for ABP

Figure 5-9: ABP scale and shape as iterations progressed.

Similar to $N_x = 3$ from Figure 5-10 a) and b) we see that when the ECG, ABP pdf's for each state are plotted together the resultant distribution approximatly matches the original ECG, ABP distributions. This can also be verified from Figure 5-11 and 5-12 where the overlapped distribution is placed side by side with the original distribution.

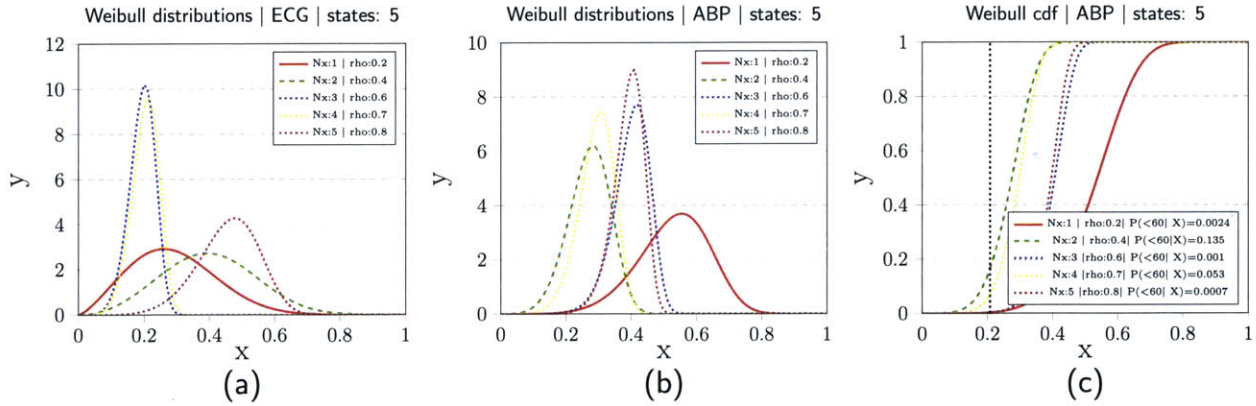


Figure 5-10: #Patients: 500 | #Sampling iterations: 1000 | ECG,ABP pdf,cdfs

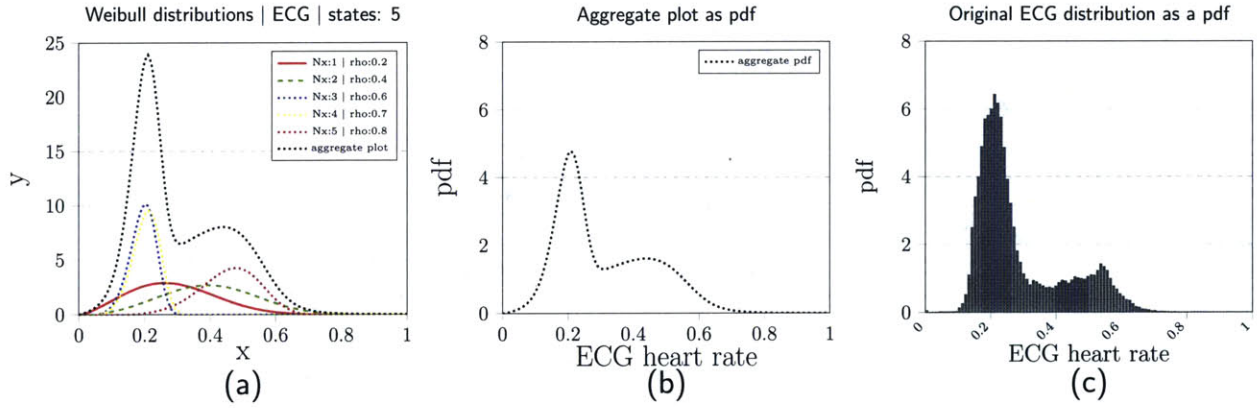


Figure 5-11: Nx =5 | Comparison of aggregate pdf vs original pdf for ECG

The state distributions for $N_X = 5$ is given in Figure 5-13. We see that there are a large number of states associated with state = 4 (yellow) which is related to low ECG and low ABP.

Table 5.10 shows the statistics related to the $scale(\lambda)$, $shape(s)$ for each state over all the Gibbs sampling rounds for $N_x = 5$.

This table was populated by recording the $scale(\lambda), shape(s)$ value for each ECG, ABP state at the end of each Gibbs iteration and calculating the statistics (standard deviation

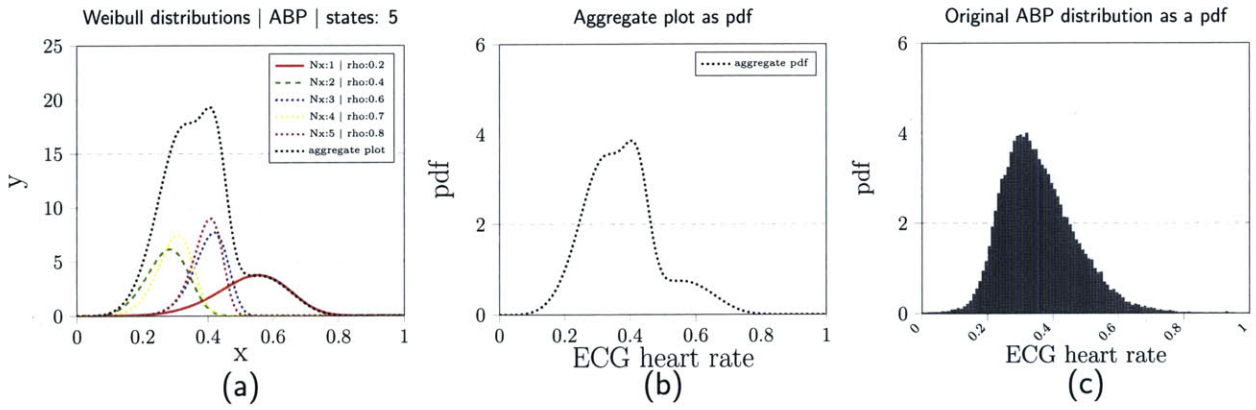


Figure 5-12: Nx =5 | Comparison of aggregate pdf vs original pdf for ABP

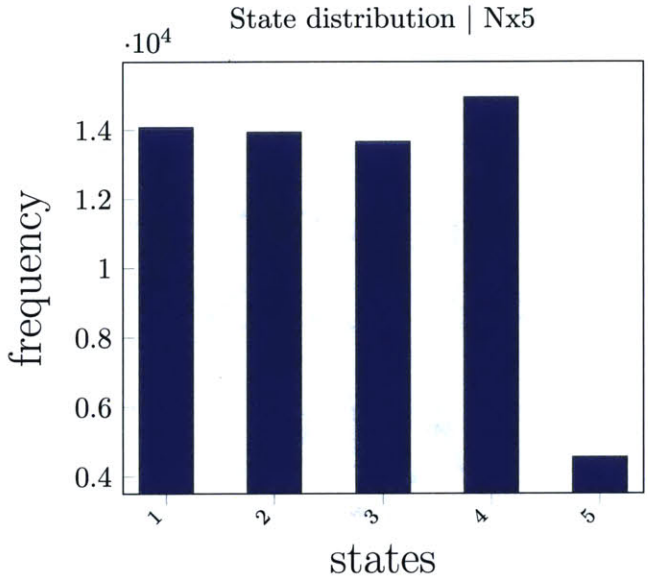


Figure 5-13: State distribution over time slices for all patients| patients = 500 | sampling iterations = 1000 | total # states = 61135

and mean) for each parameter. The entries with the higher standard deviations correspond to plots where there are large fluctuations of the parameters values during the initial Gibbs sampling iterations.

The loglikelihood fit for $N_x = 5$ is shown in figure 5-14. Again we see that as the number of sampling iterations increase the loglikelihood improves suggesting an improvement in the relative probability of observing the data given the model.

ECG ($Nx = 5$)				
	Scale		Shape	
State	mean	std.dev	mean	std.dev
1	0.344	0.032	2.478	0.267
2	0.427	0.025	2.913	0.24
3	0.211	0.003	5.589	0.511
4	0.22	0.003	5.59	0.456
5	0.508	0.015	6.942	1.553
ABP ($Nx = 5$)				
	Scale		Shape	
State	mean	std.dev	mean	std.dev
1	0.564	0.025	5.563	0.384
2	0.294	0.008	4.883	0.298
3	0.44	0.035	8.358	1.271
4	0.324	0.015	6.621	0.452
5	0.406	0.018	9.807	1.007

Table 5.10: Mean and standard deviation over all iterations for estimated parameters ($Nx = 5, Ns = 1000$)

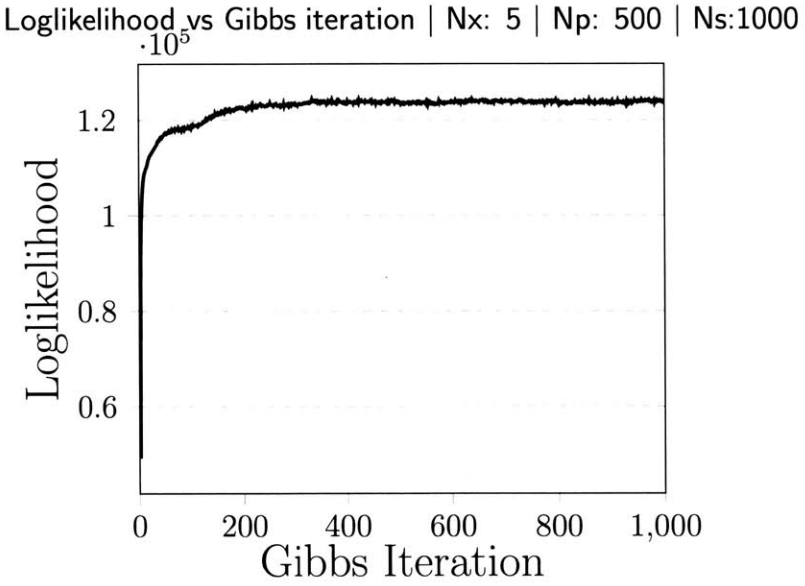


Figure 5-14: Loglikelihood vs Gibbs iteration for Nx: 5

5.2 Acute Hypotenisive Episode(AHE) event prediction

One of the main use cases of our model is to predict Acute Hypotensive Episodes (AHE). Our predictions are binary and predict whether the aggregated ABP was above or below a certain threshold. The Area Under the Receiver Operating Characteristic(AUROC) Curve is a good method to analyse the Sensitivity vs (1-Specificity) trade off of the predictions at different threshold levels. An area close to 1.00 is considered the best and an area near 0.5 is considered as an outcome of a random experiment.

Crossvalidation AUROC curves for $N_X = \{3\}$

The AUROC curves generated after 5 fold crossvalidation, testing on the withheld data is shown in Figure 5-15. The AUROC is for a $lead = 20\ min\ , lag = 120\ min$. The AUROC on the withheld data is 0.68538 for $N_x = 3$.

We observe that the AUROC of 0.685 on the withheld data set is not an extremely good value (since it's close to 0.5 which is associated with a random experiment). The reason for this might be that for $N_x = 3$ we might not be accurately modelling the original ECG, ABP distributions. The number of states might be too low to capture the complex characteristics that exist within a 500 patient data set. We also observe that fold 4 has a relatively high AUROC but since we do 5 fold cross validation we get a good idea of the variance that exists within the training data set. Results from $N_x = 3$ motivates us to experiment with larger number of states hoping to improve the AUROC.

Crossvalidation AUROC curves for $N_X = \{5\}$

For $N_x = 5$ the AUROC plots are shown in Figure 5-16. The AUROC for the withheld data is 0.79175. We observed that the prediction AUROC has improved with more states. This indicates that going from 3 states to 5 states better models the complex characteristics that lie in the ECG, ABP signals of ICU patients. We again see that the results for fold 4 is better than the others which is consistent with the observations from $N_x = 3$.

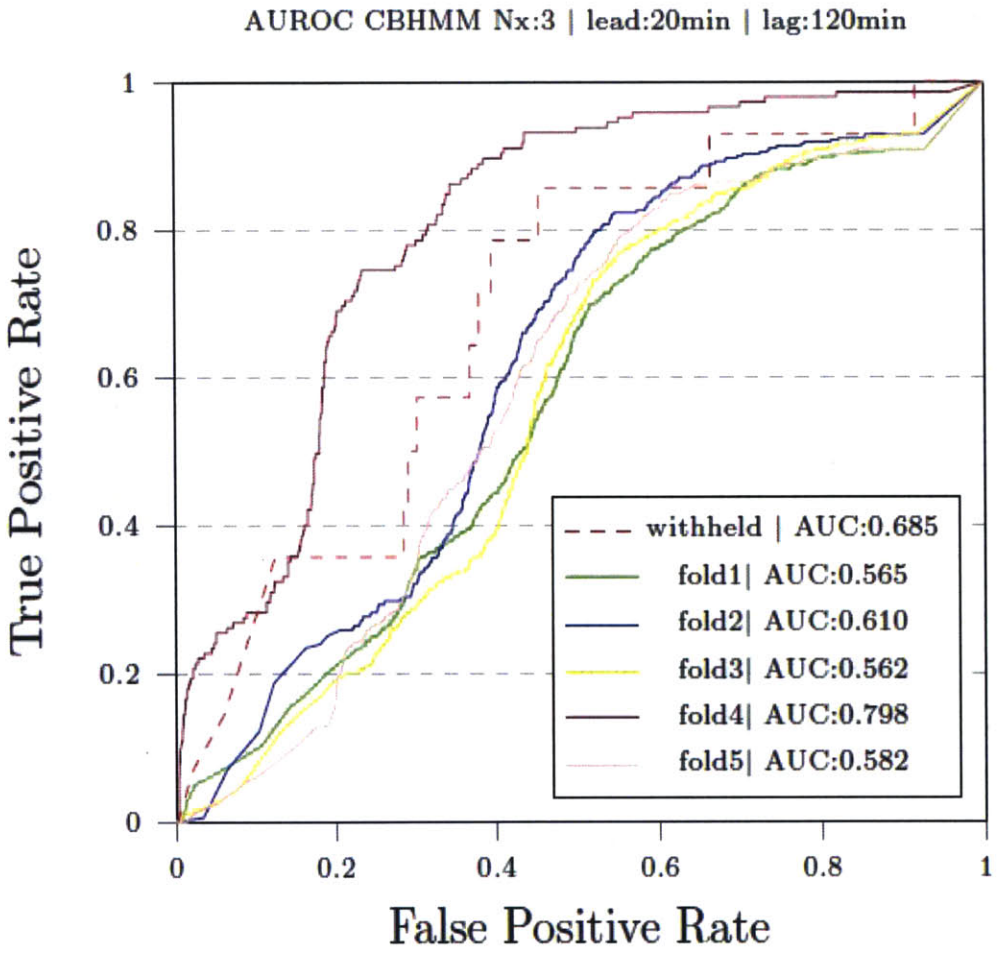


Figure 5-15: $Nx = 3$ | ROC curves for the crossvalidation folds and withheld data

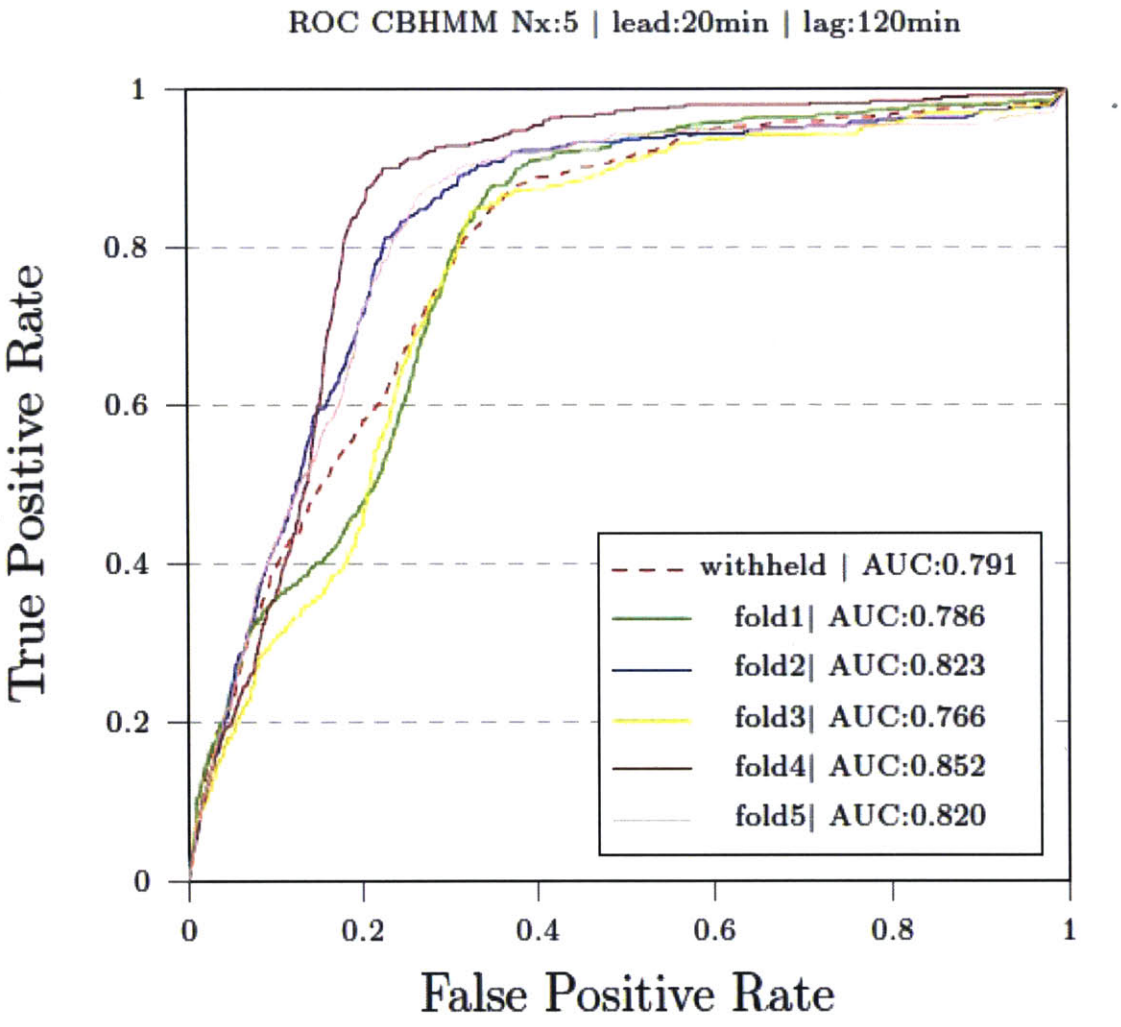


Figure 5-16: $Nx = 5$ | ROC curves for the crossvalidation folds and withheld data

Table 5.11 shows a comparison between the average AUROC for the crossvalidation folds vs withheld data. The results for the crossvalidation and withheld data are similar and this indicates that there is no huge variance between the training and testing data sets.

N_x	Average over 5 folds	Withheld
3	0.6234±.09	0.6853
5	0.8094±.03	0.7917

Table 5.11: Comparison table

5.3 Comparison of Discrete vs Latent State Space Copula HMM

Part of our exercise involves comparing the Latent State Space Copula Model results with the Classical Discrete HMM. We used the Classical Discrete HMM described in [Collazo, 2015] as our choice of discrete HMM.

The results for $N_x = \{3, 5\}$ for the discrete HMM is given in the following sections. In Figure 5-17 we show results where we ran the discrete HMM with a lead = 20 min and with lags ranging from 20 min to 120 minutes. For each lead,lag combination we ran the discrete HMM over 5 folds. The error bars in the plot are generated based on the results from the folds.

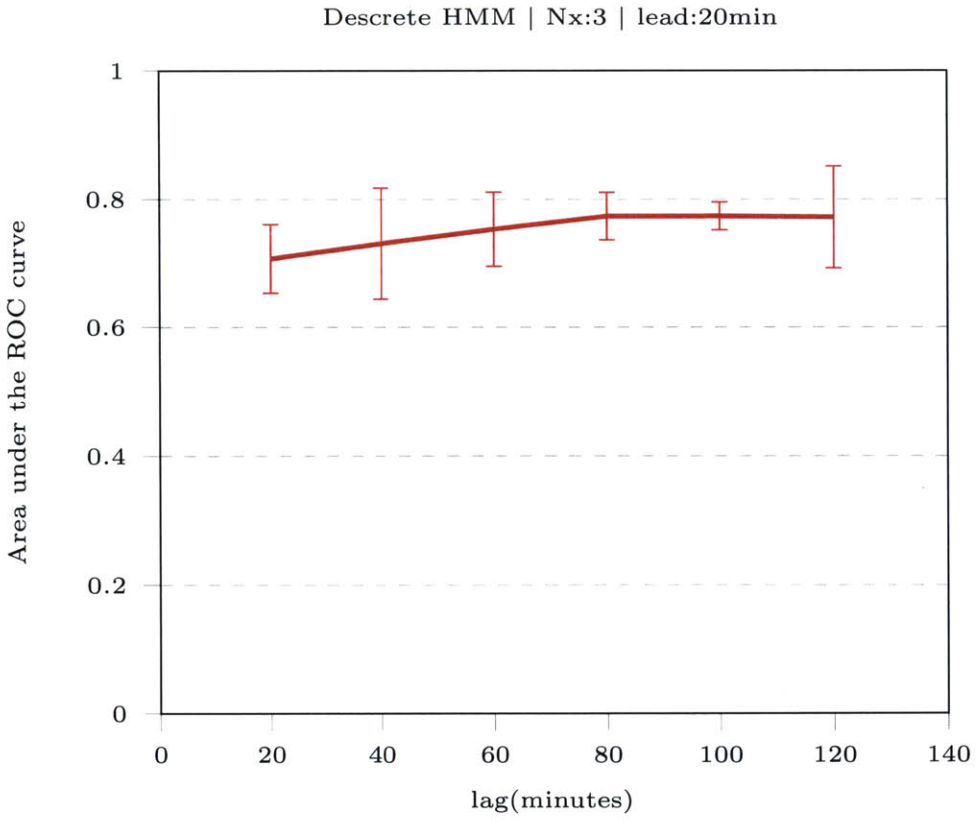


Figure 5-17: Discrete HMM AUROC for $N_x = 3$ for different lags

It's observed that beyond a certain point (80min for $N_x = 3$) increasing the lag does not

increase the AUROC significantly. In other words the previous 80 min captures most of the information that the HMM uses for prediction.

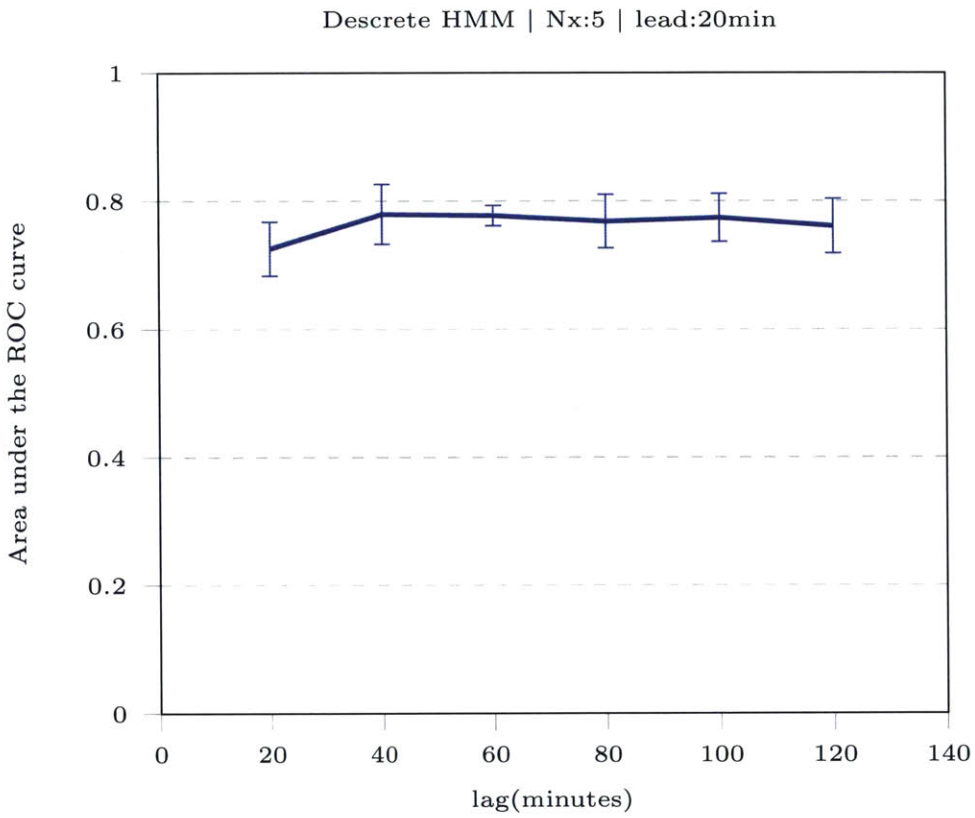


Figure 5-18: Discrete HMM AUROC for $N_x = 5$ for different lags

The results for the discrete HMM for $N_x = 5$ is given in Figure 5-16. Again for a $lead = 20\ min$ we experimented with different lags ranging from 20 min to 120 min. It's observed that beyond 40 minutes an increase in lag does not increase the prediction AUROC.

A comparison between the Latent State Space Copula Model and the Classical Discrete HMM is given in Table 5.12 below.

	AUROC for lag = 120 min	
HMM	$N_x = 3$	$N_x = 5$
Latent State Space	0.68538	0.79175
Discrete	0.74	0.7609

Table 5.12: Comparison table with results from the withheld data set

For $Nx = 3$ we compare the $lead = 20\ min$, $lag = 120\ min$ entries of the Latent State Space Copula Model and Classical Discrete HMM. The copula model has a AUROC of 0.6853 on the withheld data set. The discrete HMM has a average AUROC of 0.74. We see

that for $Nx = 3$ the Classical Discrete HMM performs better than the Latent State Space Copula Model. For $Nx = 5$ the copula model has a AUROC of 0.79175 on the withheld data set. The discrete HMM has a AUROC of 0.7609. We see that for $Nx = 5$ the Latent State Space Copula Model is better than the discrete HMM. In general going from 3 states to 5 states leads to better predictions. We cannot definitely say that increasing the number of states give us better AUROC but the structure of the HMM's gives us a framework where we can experiment with the number of states and find a suitable number that's best for the prediction problem at hand. In the context of this thesis, based on the Table 5.12 we can state that with $N_x = 5$ states, a lag of 120 minutes and for a lead of 20 minutes the Latent State Space Copula Model is the best model that can be used for blood pressure prediction.

Chapter 6

Conclusion and future work

6.1 Contribution

As mentioned in Chapter 1 the contribution of this thesis is in

1. Aligning and matching the separate ABP and ECG signals.
   This task was achieved successfully and the aligned and matched signals were used for experiments and to generate results.

2. Proposing the novel Latent State Space Copula Model for jointly modelling the ECG and ABP.
   This task was achieved and the latent space copula model provides a framework where correlated signals can be modelled and used for prediction.

3. Comparing the novel Latent State Space Copula Model with a Classical Discrete HMM.
   This task was achieved with ROC results comparison between the Latent State Space Copula Model and a Classical Discrete HMM. It was shown that the Latent State Space Copula Model for states $N_x = 5$ produces the best AUROC.

6.2 Future work

There are interesting avenues that can be explored and extensions that can be made with this research. Some are

1. **Model more than 2 correlated variables.**

   Currently only 2 variables are used in the model. But the model can actually use up to n correlated variables. The copula structure allows any number of variables to be modelled and the only barrier to overcome is the computation cost.

2. **Try other correlated variable combinations.**

   When using ECG, ABP as the example pair, the ABP signal is considered to dominate when making predictions for future blood pressure or predicting AHE events. It would be interesting to model correlated but non-dominated variables (i.e ABP and a electronic medical record time series variable).

3. **Try larger data sets.**

   In this thesis signals from 650 patients out of approximately a 6000 patient data set was used. It would be interesting to see the performance of the model for all 6000 patients and address the associated challenges when scaling to larger data sets.

4. **Try other data sets.**

   It would also be interesting to model a completely different data set and evaluate how the model performs (i.e education data, financial data).

6.3 Conclusion

In this thesis we have presented a novel Probabilistic Graphical Model structure called the Latent State Space Copula Model which provides a framework to model the joint behavior of correlated random variables. We have used it to model the correlation between the ECG, ABP signals from the data taken from the ICU. We have also used it to make predictions of AHE events for ICU patients. We hope that this research is carried forward and used in various domains to model complex relationships between variables and to make predictions that contribute to society as a whole.

Appendix A

Code

Code for model prediction, Chapter 3

```
function [prob_under_60_cumsum] = ...
    Prediction_generate_observations(y_range,z_range,Phi,...
                                rho,prediction_threshold_normalized)


  z_length=length(z_range);
  state_length=length(Phi.X1(:,1));
  cc=hsv(state_length);

  %creat a meshgrid of range of values of Y,Z and state values in
  %"obs_values_per_state"
  [y_mesh,z_mesh,s_mesh]=meshgrid(y_range,z_range,1:state_length);
  obs_values_per_state=[y_mesh(:),z_mesh(:),s_mesh(:)];
  [rows,cols]=size(obs_values_per_state);
  joint_YZ_state=zeros(rows,1);

  %compute P(Y,Z|X) for each Y,Z,X value and store it in "joint_YZ_state"
  for i=1:rows
      y=obs_values_per_state(i,1);
      z=obs_values_per_state(i,2);
      state=obs_values_per_state(i,3);
      joint_YZ_state(i)=priorObservation(y,z,state,...
```

```
21        {Phi.scale1(:,Phi.s) Phi.shape1(:,Phi.s)},
22        {Phi.scale2(:,Phi.s) Phi.shape2(:,Phi.s)},rho);
23
24    end
25
26    %append "obs_values_per_state" and "joint_YZ_state"
27     joint_YZ_matrix=[obs_values_per_state joint_YZ_state];
28
29    z_marginal=zeros(z_length,state_length);
30    z_idx=0;
31    %sum over Y to get P(Z|X)
32    for i=z_range
33        z_idx=z_idx+1;
34        for s=1:state_length
35          z_marginal(z_idx,s)=...
36          sum(joint_YZ_matrix(joint_YZ_matrix(:,2)==i & ...
              joint_YZ_matrix(:,3)==s ,4));
37
38        end
39    end
40
41     %normalize P(Z|X),
42      if sum(z_marginal(:)) ~0
43          z_marginal=z_marginal./sum(z_marginal(:));
44      end
45
46
47    figure
48    prob_under_60_cumsum=zeros(state_length,1);
49    legend_str={};
50    for i=1:state_length
51      x_values=0:.01:1;
52      %given P(Z|X) compute the CDF(Z|X)
53      cumsum_values=cumsum(z_marginal(:,i)/sum(z_marginal(:,i)));
54      plot(x_values,cumsum_values,'color',cc(i,:));
55      hold on;
```

```
56      %check where the 60mmHg point lies and get the associated
57      %P(Z<60mmHg_normalized|X)
58      prob_under_60_cumsum(i)=...
59      max(cumsum_values(x_values<prediction_threshold_normalized));
60      legend_str(i)=...
61      {['state' ,num2str(i),...
62      '|P(<60|X)=',num2str(prob_under_60_cumsum(i))]};
63
64    end
65    prob_under_60_cumsum
66    plot(ones(length(0:.1:1))*prediction_threshold_normalized,...
67                                                    0:.1:1,'k--');
68    title('cdf using cumsum')
69    legend(legend_str);
70    hold off
71
72
73    %plot the marginals
74    figure
75    for s=1:state_length
76        plot(0:.01:1,z_marginal(:,s),'color',cc(s,:));
77        hold on;
78    end
79    plot(ones(length(0:.0001:max(z_marginal(:))))*...
80         prediction_threshold_normalized,...
81         0:.0001:max(z_marginal(:)),'k--');
82    title('Z marginal')
83    legend({'state1','state2','state3'})
84    hold off
85    %compute emphirical cdf,get prob < 60mmHg
86
87  end
```

Bibliography

[Abernethy, 1996] Abernethy, R. B. (1996). *The new Weibull handbook.* RB Abernethy.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning.* springer.

[Casella and George, 1992] Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174.

[Clifford et al., 2009] Clifford, G. D., Scott, D. J., and Villarroel, M. (2009). User guide and documentation for the mimic ii database. *MIMIC-II database version*, 2.

[Collazo, 2015] Collazo, B. (2015). *Machine Learning Blocks.* PhD thesis, Massachusetts Institute of Technology.

[Cornell, 2011] Cornell (2011). Cornell university,ece1810,lab. [Online; accessed 30-August-2015].

[Cuesta Infante and Veeramachaneni, 2015] Cuesta Infante, A. and Veeramachaneni, K. (2015). Bayesian copula dependence models for time series.

[Dalla Valle, 2009] Dalla Valle, L. (2009). Bayesian copulae distributions, with application to operational risk management. *Methodology and Computing in Applied Probability*, 11(1):95–115.

[Dernoncourt, 2014] Dernoncourt, F. (2014). *BeatDB: an end-to-end approach to unveil saliencies from massive signal data sets.* PhD thesis, Massachusetts Institute of Technology.

[Dernoncourt et al.,] Dernoncourt, F., Veeramachaneni, K., and OReilly, U.-M. Gaussian process-based feature selection for wavelet parameters: Predicting acute hypotensive episodes from physiological signals.

[Dernoncourt et al., 2013] Dernoncourt, F., Veeramachaneni, K., and OReilly, U.-M. (2013). beatdb: A large scale waveform feature repository. In *NIPS Workshop on Machine Learning for Clinical Data Analysis and Healthcare.*

[Dubin, 2000] Dubin, D. (2000). *Rapid interpretation of EKG's: an interactive course.* Cover Publishing Company.

[El-Gohary and McNames, 2007] El-Gohary, M. and McNames, J. (2007). Establishing causality with whitened cross-correlation analysis. *Biomedical Engineering, IEEE Transactions on*, 54(12):2214–2222.

[Eliason, 1993] Eliason, S. R. (1993). *Maximum likelihood estimation: Logic and practice*, volume 96. Sage Publications.

[Fong et al.,] Fong, A., Mittu, R., Ratwani, R., and Reggia, J. Predicting electrocardiogram and arterial blood pressure waveforms with different echo state network architectures.

[Gelfand and Smith, 1990] Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741.

[Gopal, 2014] Gopal, V. (2014). *PhysioMiner: a scalable cloud based framework for physiological waveform mining.* PhD thesis, Massachusetts Institute of Technology.

[Kim et al., 2014] Kim, Y. B., Seo, J., and OReilly, U.-M. (2014). Large-scale methodological comparison of acute hypotensive episode forecasting using mimic2 physiological waveforms. In *Computer-Based Medical Systems (CBMS), 2014 IEEE 27th International Symposium on*, pages 319–324. IEEE.

[Li and Clifford, 2012] Li, Q. and Clifford, G. D. (2012). Signal quality and data fusion for false alarm reduction in the intensive care unit. *Journal of electrocardiology*, 45(6):596–603.

[Liu et al., 2010] Liu, C., White, R. W., and Dumais, S. (2010). Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 379–386. ACM.

[McCool, 2012] McCool, J. I. (2012). *Using the Weibull distribution: reliability, modeling and inference*, volume 950. John Wiley & Sons.

[Nelsen, 2007] Nelsen, R. B. (2007). *An introduction to copulas.* Springer Science & Business Media.

[PhysiologyWeb, 2015] PhysiologyWeb (2015). Mean arterial pressure calculator. [Online; accessed 30-August-2015].

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

[Rinne, 2008] Rinne, H. (2008). *The Weibull distribution: a handbook.* CRC Press.

[Saeed et al., 2011] Saeed, M., Villarroel, M., Reisner, A. T., Clifford, G., Lehman, L.-W., Moody, G., Heldt, T., Kyaw, T. H., Moody, B., and Mark, R. G. (2011). Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952.

[Sayadi and Shamsollahi, 2013] Sayadi, O. and Shamsollahi, M. B. (2013). Utility of a nonlinear joint dynamical framework to model a pair of coupled cardiovascular signals. *i. Utility of a nonlinear joint dynamicalframework to model a pair of coupled cardiovascular signallth Informatics, IEEE Journal of*, 17(4):881–890.

[Simpson, 1968] Simpson, D. P. (1968). Cassell's new latin dictionary.

[Singh et al., 2010] Singh, A., Tamminedi, T., Yosiphon, G., Ganguli, A., and Yadegar, J. (2010). Hidden markov models for modeling blood pressure data to predict acute hypotension. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 550–553. IEEE.

[Stell, 2015] Stell, A. (2015). Avert-it project. [Online; accessed 30-August-2015].

[Stell et al., 2009] Stell, A., Sinnott, R., Jiang, J., Donald, R., Chambers, I., Citerio, G., Enblad, P., Gregson, B., Howells, T., Kiening, K., et al. (2009). Federating distributed clinical data for the prediction of adverse hypotensive events. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1898):2679–2690.

[Sun et al., 2006] Sun, J., Reisner, A., and Mark, R. (2006). A signal abnormality index for arterial blood pressure waveforms. In *Computers in Cardiology, 2006*, pages 13–16. IEEE.

[The MathWorks, 2015] The MathWorks, I. (2015). Classification of variables in parfor-loops. [Online; accessed 30-August-2015].

[Veeramachaneni et al., 2015] Veeramachaneni, K., Cuesta-Infante, A., and O'Reilly, U.-M. (2015). Copula graphical models for wind resource estimation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2646–2654. AAAI Press.

[Waldin, 2013] Waldin, A. (2013). *Learning Blood Pressure Behavior From Large Blood Pressure Waveform Repositories and Building Predictive Models.* PhD thesis, Masters thesis, MIT.

[Wang et al., 2008] Wang, M., Rennolls, K., and Tang, S. (2008). Bivariate distribution modeling of tree diameters and heights: Dependency modeling using copulas. *Forest Science*, 54(3):284–293.

[Yartsev, 2015] Yartsev, A. (2013-2015). Normal arterial line waveforms. [Online; accessed 30-August-2015].

[Yoon, 2009] Yoon, B.-J. (2009). Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6):402.

[Zong, 1998] Zong, W. (1998). wqrs: Single-lead qrs detector based on length transform. [Online; accessed 30-August-2015].

[Zong et al., 2003] Zong, W., Heldt, T., Moody, G., and Mark, R. (2003). An open-source algorithm to detect onset of arterial blood pressure pulses. In *Computers in Cardiology, 2003*, pages 259–262. IEEE.

[Zong et al., 2004] Zong, W., Moody, G., and Mark, R. (2004). Reduction of false arterial blood pressure alarms using signal quality assessement and relationships between the electrocardiogram and arterial blood pressure. *Medical and Biological Engineering and Computing*, 42(5):698–706.