Learning Blood Pressure Behavior from Large Physiological Waveform Repositories

Alexander Waldin WALDINA@MIT.EDU
Kalyan Veeramachaneni KALYAN@CSAIL.MIT.EDU
Una-May O'Reilly UNAMAY@CSAIL.MIT.EDU
Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139 USA

Abstract

We consider the problem of predicting an ICU patient's future blood pressure from a recording of his recent blood pressure history. We envision a system that that can use varying amounts of a patient's blood pressure history and that can predict at varying times in the future. We model and train a temporal Bayesian network on hundreds of patients extracted from a large repository of physiological waveforms and use it to make predictions up to 2 hours into the future.

1. Introduction

Patients with serious injuries, post-surgery trauma or unstable health conditions are usually transferred into a hospitals Intensive Care Unit (ICU). Within this clinical setting, health care professionals closely monitor them using bedside sensor systems that continuously record signals such as electrocardiogram and arterial blood pressure waveforms. The ICU dataset we reference in this paper comprises approximately 18 different sensor signals recorded at a much higher rate (125 samples/second) than the required Nyquist rate for these signals (4 samples/sec).

Such high **frequency** information is primarily used to generate alarms (Zong et al., 2004). Alarms are reactions to the current state of the patient rather than predictions. In ICU settings, predictions about a possible future state of a patient are usually made through subjective assessments by health care professionals who summarize patient status at intermittent, relatively lengthly (on the order of hours) intervals.

Our interest is in, as yet, untapped predictive information within such continuous waveforms. Can these waveforms be mined for patterns and can they generate useful insights and predictions about the future state of the patient? Pattern recognition and prediction will require adequate amounts of data. After 10 years of waveform archiving, the MIMIC-II database (Saeed et al., 2002; 2011) which curates ICU data from 5 different ICUs at Bostons Beth Israel Deaconess Hospital contains continuous waveforms for more than 15000 adult patients.

We have started to mine ABP waveform data by formulating a flexibly defined prediction problem which relies solely on the waveform data. We try to predict the future value of an event statistic which is defined over a parameterized time duration.

Apart from the continuous waveforms, rich clinical information about the medications, tests and subjective assessments are available for these patients (Saeed et al., 2002; 2011). This rich insightful information has been mined for a variety of clinical retrospective studies and knowledge discovery. In this paper, we focus solely on the waveforms in order to address a fundamental research question: How much information is in these waveforms alone? We attribute the success in prediction of a future state of the patient as the richness in the information content in the waveform. Additionally, if we succeed with this question, the implications are broader because such a predictive system only relies on a hardware and a software system allowing its deployability in a wide variety of situations including underdeveloped countries and developing countries.

We hypothesize that we will be able to build an accurate prediction system because there is a lot of data. This provides sufficient instances of specific conditions like post-surgical trauma, disease, and patient health states. We ask how well modeling the temporal/longitudinal data along with realization of existence of multiple distinct patterns which affect a future state, can capture, with high fidelity, predictive information. We propose to use a temporal Bayesian

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

network (TBN) with hidden state variables.

We encounter a number of challenges in being able to use the repository effectively and also in the development of the temporal Bayesian network based model. We present the strategies we employed to overcome these and present results on a set of patients. We proceed as follows. We present the problem formulation in Section 2. We present related work in section 3. In section 4 we present the details about the dataset and the challenges encountered. In section 5 we present, how we assemble a challenging dataset for learning. In section 6 we present the temporal Bayes net model and present the results in section 7. Section 8 concludes.

2. Problem Formulation

In our system, given a continuous mean arterial signal derived from the waveform recording, $m(1 \ldots t)$, our goal is to predict the value of a statistic defined for the sample over a time period $t+\tau$-to- $t+\tau+\alpha$. τ is called the *lead* time, α is called the *prediction* window. The algorithm is allowed to use any amount of historic values of the signal $m(1 \ldots t)$. However, one can bound this time period by, γ, and it is called *lag*. The problem formulation is visually depicted in Figure 3.

Our goal is to allow the users to specify any *lead* or *lag* time based on their requirements. We envision that different ICUs, different patients and different situations have different needs that would drive the values for these parameters. Additionally, we also envision that the algorithm should allow the ability to define the *prediction* window size and the definition of the *statistic* derived over this *prediction* window. This flexibility allows personalization and ability to use our system under different situations.

3. Related Work

Initial research on utilizing continuous waveforms at the bed side has focused on reducing false alarm rates by utilizing information from two or more waveforms (Zong et al., 2004). Use of other waveforms allow the system to reduce the noise in the waveform from which an alarm is generated. Additionally, values from one waveform can be utilized to impute missing values in other waveforms(Moody, 2010). This research enhances the alarm generating systems by reducing noise in a waveform recording, reducing false alarms and imputing missing values.

Another line of research focusses on predicting, with a specified lead, an acute hypotensive event. An AHE event was said to occur if 90% if the mean arterial pressure during a prediction window of 30 minutes was $< 60mmhg$. The data corresponding to 60 patients, together with the problem definition, was released as a competition in Computers and Cardiology conference in 2009 (Moody & Lehman, 2009). The problem was then defined as a binary classification problem and a number of researchers developed methods to solve this problem (Henriques & Rocha, 2009; Mneimneh & Povinelli, 2009). Acute hypotension, as defined in this competition, was an extreme event. In contrast, in this paper, our focus is to predict the state mean arterial pressure defined by discrete bins defined over its entire range, with a parameterized *lead* window. Flexible definition of an event and a *lead* accommodates the specific situation of the patient but fundamentally challenges the learning algorithm. They require greater precision in prediction and yield data with overlap across different states.

Knowledge discovery in electrocardiogram signals have been the focus of (Saria et al., 2010a; Syed et al., 2007). (Saria et al., 2010a;b) developed a time series topic model that relied on latent variables to model heart rate and respiratory rate for premature infants in the neonatal ICU. The goal was to create an accurate representation of the data, paying attention to domain specific nuances and then perform supervised learning against an outcome of interest. The outcome of interest was the state/condition of the infant when they are released as marked by the doctors. Similarly, (Syed et al., 2010) attempt to identify motifs in electrocardiogram data. They first reduce the physiological waveform to finite discrete symbols and then attempt to find patterns that may be associated with sudden cardiac death. Both these systems are, perhaps closely related to what we are attempting to do, but do not define their problem as the one of predicting the future state of the signal itself. In this paper, we challenge ourselves to be able to predict the future state of the signal itself with minimal information from doctors or clinical information.

4. Raw ABP waveforms and Challenges

The entire MIMIC II version 3 waveform database includes 23,180 sets of recordings and over 3 TB of compressed data in all (Saeed et al., 2002; 2011). In this paper, we consider only the arterial blood pressure (ABP) waveform record. We propose this data contains information which will allow us to infer how a patient's blood pressure will develop in the future. Of all of the patients in the MIMIC II $v3$ database, 6232 patients have ABP waveform records. Although

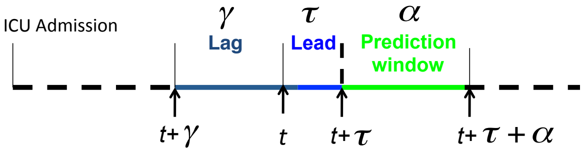

Figure 1. Predicting a future value of blood pressure. The prediction system has three parameters, the *lead*, *lag*, and the prediction window.

the number of patients may seem small, two additional factors make this data quite large. First, for each patient data is stored for many hours. Second, the signals were sampled at 125Hz. Hence the total data from the blood pressure signal alone adds up to 2 Terabytes worth of decompressed data. Over 240,000 hours worth of data has been recorded. Before we can build a predictive model for our problem described in Section 2 we process and validate this data on a patient-by-patient basis.

Processing:The ABP signal is an oscillatory waveform that repeats with a period known as the *beat duration* for a patient. This period varies not only from patient to patient, but also within an individual patient's signal. In a single beat, the blood pressure rises from a low value to a peak value which is called a *systole* and the pressure value is called systolic pressure (P_s). This phase of the signal is called *anacrotic limb*. The value slowly starts to decline and has a bump (a small rise and then fall) called *diacrotic notch* and finally falls to the lowest value known as diastolic pressure (P_d). This second phase of the beat is called *dicrotic* limb. Fundamentally, any complex physiological changes within a patient might be captured in this single beat and could repeat for multiple beats over a duration of time. Capturing these signatures at beat level and identifying a persistent pattern and correlating it with a future condition is the focus of this paper. Hence it is paramount for us to process patient data beat-by-beat. However, the above said variation in beat duration introduces a challenge in isolating samples that correspond to a beat. To isolate samples that correspond to a single beat, we first apply the beat onset detection (Zong et al., 2003). Given contiguous samples, the beat onset detection algorithm returns the start and the stop sample index for each beat.

Validation: After tagging the signal with start and stop, each beat is then validated based on the criteria specified in (Sun et al., 2006). We now tag the beat with a validity index $v \in (0, 1)$. 6 different conditions were applied to the data corresponding to a beat to check whether or not the beat is valid. These conditions are: $P_s \leq 300mmHg$, $P_d > 20mmHg$, $30 \leq P_m \leq 200mmHg$, $20 \leq f \leq 200$, $P_s - P_d \geq 20mmHg$, where P_m is the mean arterial pressure, f is the hear rate given by $60/T$ where T is the beat duration. In addition, three conditions compare adjacent beats and impose that two subsequent beats not have a difference in systolic or diastolic pressure greater than 20mmHg and the difference in beat duration be greater than $\frac{2}{3}$ of a second.

Identifying contiguous segments Our goal in this paper is forecasting (see Section 2). Thus we have to assemble data that is contiguous in time to be able to learn from it. For the purpose of this paper, we will call a contiguous time signal a *segment*. It is important to note that the data per patient , acquired from MIMIC II $v3$, is not a single continuous signal, instead the signal is broken up into segments. The breaks that indicate the beginning of a new segment, due to causes such as readjustment of the measurement devices, are indicated by gaps in the time axis. In order to detect segment breaks we consider a simple rule: whenever two consecutive beats are more than 700 samples or 5.6 seconds apart, we declare that there is a break in the signal and we consider the signal after the break as a new segment.

We introduce a second set of breaks after tagging the beats with the validity index. We think an algorithm can tolerate a small number of contiguous beats declared as invalid, but if a large set of contiguous beats are declared invalid we have to break the signal apart into two different segments. After this step for each patient we have contiguous segments of disparate lengths.

Cloud based architecture for processing: We employed 100 nodes on our internal cloud to process and validate the beats for all 6232 patients. The architecture used was similar to map-reduce with a master server with list of jobs, where each job corresponded to a processing one patient's data and slaves executed the jobs. It took us approximately 24 hours to finish the process of validating all the beats.

5. Extracting a challenging dataset for predictive modeling

Our goal is to turn the MIMIC beat dataset into an experimental dataset, from which we can learn a temporal bayesian network (TBN). We do this in four steps, first we select a subset of patients we want to use for learning and testing. Second, we define a learning-segment, which encompasses the contiguous time series over which a DBN is learnt. Third, we define the concept of a unit-time segment and define aggregate features that constitute the random variables in our TBN. Fourth, we quantize the random variables into discrete values. Finally, we present a strategy to select segments that have interesting state-transition patterns.

Step 1. Selecting a subset of eventful patients In order to find patients that could provide us eventful data to learn from, patients were sorted by how many of their beats had an average blood pressure below 60 mmHg. From all of these patients, 1000 patients with the largest number of such beats were selected as the data sources from which we construct our learning dataset. The reason for selecting patients with a large number of beats with an ABP below 60mmHG was that these were patients who were more likely to have hypotensive events, that is, a sudden drop in blood pressure.

Step 2: Extracting and selecting the segments for these patients

We are interested in predicting at most 2 hours into the future and in providing at most 2 hours of data as evidence. This means that it is necessary to have data from patients that have at least one continuous four hour segment of data relatively free of large portions of contiguous invalid beats. We refer to such a four hour segment as a learning-segment.

Step 3. Feature extraction, defining a unit time and creating aggregate features

As we mentioned previously, each beat corresponds to a complex physiological cycle and our goal is to capture a feature within a beat period and its pattern over time. In this paper, we start with the mean arterial pressure, defined as $\frac{2*P_d+P_s}{3}$. Given a patient's segment wise data, we first extract the feature value for every valid beat in the segment. We then decide upon a fundamental unit of time, β per which different summary statistics, aggregate summaries of this feature are extracted. Given β we split the beats in a learning-segments into multiple sets of beats whose total duration adds up to β. In this paper, $\beta = 20 mins$. This creates $L = 12$ unit-time sub-segments within the 4 hour learning segment. Within each unit-time sub segment we place a requirement that it should have at least 50% beats as valid. We apply this condition and remove the learning-segments that do not meet this criterion.

We note that each unit-time sub-segment has a large number of beats and the number of beats varies from one sub-segment to the next. It is hence not possible to use the value of the raw mean arterial pressure as the random variables within a DBNT. Instead we define aggregate features that summarize such a sub-segment.

Aggregate features For the raw MAP values for a set of beats in every consecutive 20 minute sub-segment within the 4 hour learning segments we create the following summaries of the sub-segment:

Moments: We evaluate the mean m, standard deviation, sd, kurtosis, ku, skew, sk of the MAP values.

Trends: We model the MAP with respect to time with linear regression and take the slope to represent the trend in the signal. This is denoted by tr.

Differentials: We calculate the rate of change of the signal (velocity) $\frac{dX}{dt}$ and add its mean value as another feature and mean value rate of change of velocity (acceleration) $\frac{d^2X}{dt^2}$ as another feature. These are denoted as v, a.

Thus we summarize each sub-segment with these 7 aggregate features and these form the observable set for each time step in the TBN.

Step 4. Quantizing Aggregate-Features The joint distribution in our DBN is multinomial for discrete valued random variables. Hence we must discretize the continuous valued features. We quantize m using the following bin boundaries: $\{55, 60, 65, 70, 75, 80, 85, 90, 95\}$ and associate a label from 1-10 with each bin. This is the value we want to predict. For each of the rest we discretize them into 10 bins as well. We collect the samples for a feature from all of the sub-segments that are provided to train the DBN and then calculate the mean and standard deviation for each of these aggregate-features. We then set two boundaries at $\mu - 2\sigma$ and $\mu + 2\sigma$ and we split the range between these two boundaries evenly into 8 bins, giving us a total of 10 bins.

Step 5. Selecting segments from these patients Depending on the problem setting, it is worth considering from which learning segments data should used. In our case, we are particularly interested in predicting blood pressure when it is volatile and when the patient is at risk of having a sudden drop in the blood pressure. This criteria also makes it a more difficult prediction problem. To identify the learning segments that have such characteristics, we define a metric called *volatility* index and measure each learning segment's volatility index. The *volatility* index for any learning segment is given by:

$$\omega_1 \frac{\sum_{t=2}^{L} |m_t - m_{t-1}|}{\sum_{l=1}^{L} |m| - 1} + \omega_2 \frac{\sum_{t=1}^{L} 1 - u(m_t - 3)}{L}, \qquad (1)$$

where u is a unit step function. The index is a weighted sum of two components. The first component evaluates the difference between two consecutive mean of MAP values in the learning segment and sums then up. The maximum value for this sum is when all two consecutive mean values are 9 states apart, considering the feature m has 10 states. This maximum value is used to normalize the sum. The second component in the index simply identifies the number of sub segments whose MAP state is ≤ 3. Again it is normalized by the maximum possible value for this number. The two components are weighted with ω_1 and ω_2 and $\omega_1 + \omega_2 = 3$. Hence the volatility index $v \in \{0, 3\}$. For example, if the 12 aggregate-mean values from the sub segments that make up an experimental segment are $\{2, 5, 7, 3, 5, 1, 6, 3, 4, 3, 2, 3\}$ then the *volatility* index is calculated as:

$$|5-2| + |7-5| + |3-7| + |5-3| + |1-5| + |6-1| + |3-6| + |4-3| + |2-3| + |3-2| / (11 * 9) + 2 * 3/12 = 0.7727$$

For each patient we rank his learning-segments by their volatility index, and in order to create our learning dataset, we choose from each patient exactly one learning-segment, which has the greatest *volatility* index. The reason for this is that we do not want to create a dataset that is biased toward any particular patient as some patients have many more learning segments than others. As a result we obtained 855 *learning* segments. We selected 700 segments for training and 155 segments for testing.

6. Temporal Bayesian Network Model

We posit a temporal bayesian network for this problem as shown in Figure 2. This temporal bayesian network is a directed graphical model in which the patient state at any time t is represented by a s_t where $t \in \{1, L\}$. L is the total number of time slices (sub segments). In our network, the value of the node s_{t+1} at time $t+1$ depends only on the state of the patient at time t (markovian assumption). That is, there is only a single parent for the node s_{t+1}, given by $Pa_{s_{t+1}}$ and that is s_t. The state of the DBN is hidden and the values are inferred from the observed nodes $O_t = \{m_t, sd_t, ku_t, sk_t, tr_t, v_t, a_t\}$. The state node s_t is the parent for the observed nodes O_t at time t. One additional assumption in our model is that, given the value for the hidden state, the observed nodes are independent. This assumption could be revisited and a structure within each time slice could be learnt. Given the directed acyclic graph within a time slice, as in Figure 3 and the structure of DBN in Figure 2, we attempt to answer two questions: How many hidden states can explain our data? And, are all features necessary to build this model? We attempt to answer these two questions via cross validation.

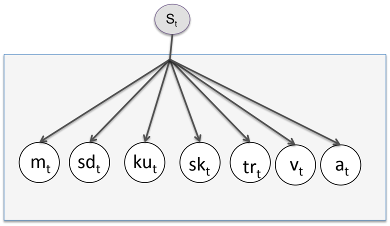

Figure 3. The directed acyclic graph within a time slice of a temporal Bayesian network.

Why Use a temporal Bayesian Network: A number of properties of this problem have motivated us to investigate a temporal Bayesian network approach. First and foremost, the temporal nature of the data calls for a model that can capture the temporal dynamics. Second, as we noticed in the previous section, the sizes of segments vary greatly and the DBN approach allows us to train a model from segments of disparate sizes. Third, modeling of the hidden state allows us to separate multiple distinct patterns in the data thus automatically grouping similar patterns for better prediction. Finally, and the most important aspect, is that it allows us to flexibly define the lead and lag during prediction. When supervised learning is employed to solve this problem, one assembles data by specifying a lead and lag time and defining the feature set for a training case and the label accordingly. If the lead and the lag are changed the model has to be re-trained.

Learning: To be able learn the parameters of the DBN we utilize an expectation maximization (EM) algorithm. The total number of parameters are $\sum_i |s| \times |f_i| + |s|$ for the first slice and $(\sum_i |s| \times |f_i| + |s|^2) \times L$, where f_i is the i^{th} feature selected. We initialize the parameters with random values and set the maximum number of EM iterations to 100. Alternatively the stopping criteria is set such that if the log likelihood does not improve by more than 0.001 the algorithm is deemed to have converged.

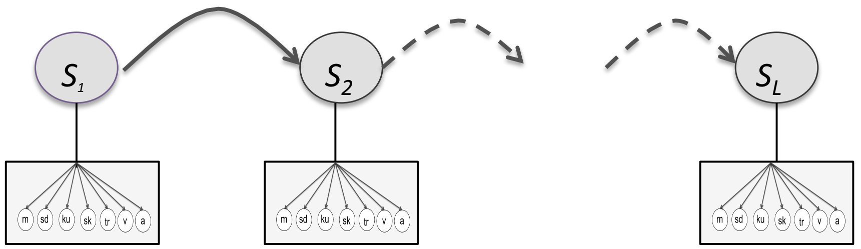

Figure 2. Temporal bayesian network for prediction of a future blood pressure state. The state transition model.

Inference and Evaluation: For each learning-segment we use, we proceed in the following way: We split it in half and provide the values for the observed nodes for first six sub-segments (2 hours) as evidence to the DBN. We then infer the corresponding hidden states. We then calculate the marginal probabilities for the variable m for each of the next 6 sub segments. We choose the most likely value (corresponding to the highest marginal probability) for m for the next 6 sub segments. This is equivalent to varying the lead time by 0, 20, 40, 60 , 80 and 100 minutes. To compare the 6 predicted observations with the 6 actual observations, we follow the following procedure, for each observation we calculate how far it is off from the actual value by evaluating $|\hat{m_l} - m_l|$ where $l \in \{0, 20, 40, 60, 80, 100\}$. This gives us six errors for each learning segment e_1, e_2, ... e_6 each corresponding to predicting the next 6 sub segments. We define $e_t = \sum_{i=1}^{6}, e_i$. We average the error across all the segments and call that the total error, E. When we attempt to do cross validation we evaluate E for each fold and then average the value for all the folds and attempt to minimize that.

Sequential feature selection: We initially set the number of hidden states as 23, then use forward feature selection to determine the number of features we should use. The forward selection process starts with the mean arterial pressure as the first feature. It then adds one feature at a time from the rest of the 6 features, performs 3-fold cross validation, each time training on the 2 sub parts of the data and testing on the third. As a result 6 different error values are evaluated and the best pair of features is selected. We examine whether the cross validation error is reduced. If the cross validation error is reduced we accept the feature and then proceed to examine triplets. This process is stopped when adding an additional feature does not produce an improvement in error. We show the cross validation error as we add features in Figure 4. After one step of this process, kurtosis ku was added as a feature and the error reduced from 10.4 to 9. After that we build DBNs for every 3 tuple and identify the best based on the cross validation error. However, we notice that the error slightly increases. As a result we stop and do not explore any more addition of features. At the end of this process, two features were selected and they were m, ku.

Number of hidden states selection: Next we determine the number of hidden states. We use the features selected through the process above. We change the hidden states from 2-35 each time learning via cross validation 3 different DBNs and evaluating the error based on the 3 folds. Figure 4 shows the cross validation error for each of the hidden state quantities. The cross validation error monotonically reduces as we increase the number of states to 7 but then behaves in an unpredictable fashion. Based on this we choose the number of hidden states as 7. Thus we build the final model with 7 hidden states, all 700 segments and m and ku as features.

A note on computational expense: During this process to determine different parameters for the DBN we built multiple DBNs and evaluated them. In the worst case scenario, the total number of DBNs learnt in the process of feature selection is $(f-1)! \times k$ where f is the number of features and k is the number of folds in k-fold cross validation. The number of DBNs learnt in selecting the hidden states is equivalent to $h-1 \times k$, where h is the maximum limit on the number of hidden states. In our experiments we learnt a total of 192 DBNs. Since cross validation is embarrassingly parallelizable we employed a machine with 24 vCPUs and 46GB shared memory and used MATLAB's parallel computation toolbox to parallelize multiple cross validation experiments.

7. Results and discussion

We use 155 learning segments as our test data and evaluate the prediction error E defined in the previous section. We evaluate the error for different lead times. Figure 5 shows the performance of the model as we vary the lead times. As a baseline we show the

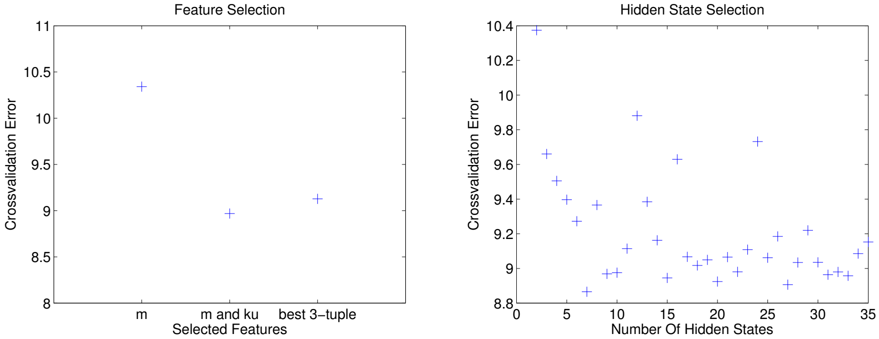

Figure 4. Cross validation error as different feature sets are selected and different number of hidden states are selected. Demonstration of feature selection (Left). Demonstration of hidden state selection (Right).

prediction based on persistence, where the future state of blood pressure is chosen to be the same as the current. We notice from this figure that for each of the lead times the model provides better performance than persistence. In addition we observe a peculiar behavior that the performance of the model reduces as we increase the lead time and performs worst at 60 mins lead time, but eventually recovers gets better at a lead time of 100 minutes.

We then vary the amount of evidence given to the DBN. For the same lead time, we gave evidence worth 120, 100, 80, 60, 40, 20 minutes. Figure 5 (lag) shows the prediction accuracy as we changed the amount of information we put in as evidence. We notice that in general less historical (lag) data did not have significant effect on the performance. The box plots have very low variance.

8. Conclusions

In this paper, we developed a Dynamic Bayesian network based approach for prediction of the blood pressure in an ICU setting. Our goal is to use the waveforms data exclusively. This entailed a number of detailed steps to process, clean, validate and form a dataset from the publicly available large arterial blood pressure waveform dataset. We built an end-to-end system and are able to choose different parameters for the TBN via cross validation. We extracted a very challenging subset of the data from the waveform dataset, presented the criteria for this selection and the results of prediction via the learnt TBN. There are number of avenues we are proceeding with regards to this research.

More features: First and foremost we expect the performance of the system will further improve as add more features and their aggregates derived from the beat level. These include domain specific features like area-under-systole, duration-of-systole, and duration-of-diastole; frequency domain features, and joint time-frequency transforms via wavelets and short-time fourier.

Defining a decision rule: Our current decision rule in deciding the blood pressure state simply selects the state that corresponds to the maximum marginal probability. We would like to modify this decision rule either by incorporating weights for different errors, or specifying a Neymen-pearson criterion (Landgrebe & Duin, 2005) that allows us to emphasize achieving higher accuracy in prediction of certain states. This will result in more domain specific model.

Learning structure: With addition of new features, we will investigate whether or not a naive structure is enough or if we can learn a structure thus giving us a better model and prediction accuracy.

Incorporating domain knowledge: We will investigate if we can incorporate domain knowledge to form a DBN that will best represent the patient transitions in an ICU.

Large scale simulations: As we add more features, we will also explore adding more segments and more patients. This will increase the size of data and the computation time of the EM algorithm. We will build a system that will exploit multiple levels of parallelism. At the lowest level it will distributed the EM algorithm and the next level it will parallelize the different cross validation experiments.

References

Henriques, JH and Rocha, TR. Prediction of acute hypotensive episodes using neural network multi-models. In *Computers in Cardiology, 2009*, pp. 549–

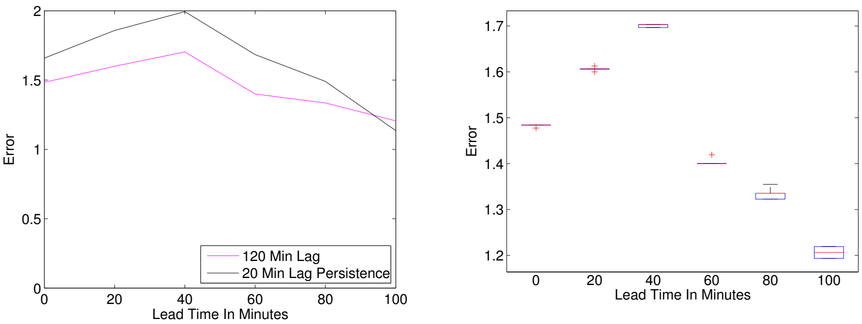

Figure 5. Prediction accuracies as a function of different lead and lags. Prediction accuracy as a function of lead time. (Left). Variation in the prediction accuracy as a function of lead times and as we vary the amount of historical data used (lag). (right)

552. IEEE, 2009.

Landgrebe, Thomas and Duin, R. On neyman-pearson optimisation for multiclass classifiers. In *Sixteenth Annual Symposium of the Pattern Recognition Assoc. of South Africa*, 2005.

Mneimneh, MA and Povinelli, RJ. A rule-based approach for the prediction of acute hypotensive episodes. In *Computers in Cardiology, 2009*, pp. 557–560. IEEE, 2009.

Moody, GB and Lehman, LH. Predicting acute hypotensive episodes: The 10th annual physionet/computers in cardiology challenge. In *Computers in Cardiology, 2009*, pp. 541–544. IEEE, 2009.

Moody, George B. The physionet/computing in cardiology challenge 2010: Mind the gap. In *Computing in Cardiology, 2010*, pp. 305–308. IEEE, 2010.

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

Saeed, Mohammed, Villarroel, Mauricio, Reisner, Andrew T, Clifford, Gari, Lehman, Li-Wei, Moody, George, Heldt, Thomas, Kyaw, Tin H, Moody, Benjamin, and Mark, Roger G. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

Saria, Suchi, Koller, Daphne, and Penn, Anna. Learning individual and population level traits from clinical temporal data. In *the Predictive Models in Personalized Medicine Workshop. Twenty-Fourth Annual Conference on Neural Information Processing Systems*, 2010a.

Saria, Suchi, Rajani, Anand K, Gould, Jeffrey, Koller, Daphne, Penn, Anna A, et al. Integration of early physiological responses predicts later illness severity in preterm infants. *Science translational medicine*, 2(48):48ra65–48ra65, 2010b.

Sun, JX, Reisner, AT, and Mark, RG. A signal abnormality index for arterial blood pressure waveforms. In *Computers in Cardiology, 2006*, pp. 13–16. IEEE, 2006.

Syed, Zeeshan, Guttag, John, and Stultz, Collin. Clustering and symbolic analysis of cardiovascular signals: discovery and visualization of medically relevant patterns in long-term data using limited prior knowledge. *EURASIP Journal on Applied Signal Processing*, 2007(1):97–97, 2007.

Syed, Zeeshan, Stultz, Collin, Kellis, Manolis, Indyk, Piotr, and Guttag, John. Motif discovery in physiological datasets: A methodology for inferring predictive elements. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):2, 2010.

Zong, W, Heldt, T, Moody, GB, and Mark, RG. An open-source algorithm to detect onset of arterial blood pressure pulses. In *Computers in Cardiology, 2003*, pp. 259–262. IEEE, 2003.

Zong, W, Moody, GB, and Mark, RG. 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, 2004.