Dynamically Modeling Patient's Health State from Electronic Medical Records: A Time Series Approach

Karla Caballero
School of Engineering
University of California, Santa Cruz
Santa Cruz, CA, USA
karla@soe.ucsc.edu

Ram Akella *
School of Information
University of California, Berkeley
Berkeley, CA, USA
akella@ischool.berkeley.edu

ABSTRACT

In this paper, we present a method to dynamically estimate the probability of mortality inside the Intensive Care Unit (ICU) by combining heterogeneous data. We propose a method based on Generalized Linear Dynamic Models that models the probability of mortality as a latent state that evolves over time. This framework allows us to combine different types of features (lab results, vital signs readings, doctor and nurse notes, etc) into a single state, which is updated each time new patient data is observed. In addition, we include the use of text features, based on medical noun phrase extraction and Statistical Topic Models. These features provide context about the patient that cannot be captured when only numerical features are used. We fill out the missing values using a Regularized Expectation Maximization based method assuming temporal data. We test our proposed approach using 15,000 Electronic Medical Records (EMRs) obtained from the MIMIC II public dataset. Experimental results show that the proposed model allows us to detect an increase in the probability of mortality before it occurs. We report an AUC 0.8657. Our proposed model clearly outperforms other methods of the literature in terms of sensitivity with 0.7885 compared to 0.6559 of Naive Bayes and F-score with 0.5929 compared to 0.4662 of Apache III score after 24 hours.

CCS Concepts

•Mathematics of computing → Time series analysis; •Applied computing → Health care information systems; •Information systems → *Content analysis and feature selection;*

Keywords

Mortality Prediction, Dynamic Linear Models, Text Mining

*and University. California, Santa Cruz, akella@soe.ucsc.edu

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783289

1. INTRODUCTION

Currently the accurate and opportune prediction of an increase in the probability of mortality is of great interest to physicians. Tools that estimate this probability aid physicians to determine the possible treatments that improve patient's health. In addition, the timely and accurate estimation of patient's probability of mortality allows us to successfully trigger a medical alarm. This estimation also permits the early identification of patients with elevated clinical risk. As a result health care providers can differentiate those patients from the ones who are stable and improving in order to assign medical resources more effectively.

Prevailing medical practice relies on frameworks such as the Apache III [13], and SAPS II [12] scores. Both methods, widely used to predict patient mortality in the Intensive Care Unit (ICU), incorporate temporal information in a limited way by only choosing the worst-case scenario values during the first 24-hour window that a patient is inside the ICU. As a result, they often overestimate the probability of mortality. Moreover, these scores are only estimated once during all the patient's stay in the ICU, which may not indicate whether the patient would recover and be discharged from the ICU in the future.

Data Mining techniques have been previously used to address in the problem of estimating the patient's mortality [15, 2, 11, 10]. Most of the existing solutions rely on training a static classifier with a patient's observed feature vector. These features can be either static, such as lab reports produced at a given point of time, or dynamic, such as waveforms of vital signs that are discretized into static features. Lee *et al.* in [15] select the last observed value from the patient's stay during the first 48 hours to fit a logistic regression model to predict the probability of mortality. However, this system cannot predict any alarm prior to the completion of the 48 hour window. This wait limits the possible treatments that can be applied to the patient. Batal *et al.* include the dynamic information by collapsing the time series of features, such as blood pressure and heart rate, into static features that are later used in a classification framework [2]. Consequently, this model does not take into account the evolution of the patient in time (i.e. the algorithm may not be able to discriminate whether an increase in the blood pressure implies a positive signal or not). Hug and Szolovits aggregate dynamic information using two frameworks: Real-time and Daily reading processing that are later incorporated into a static classifier [11].

Furthermore, most of the methods mentioned above assume the availability of all the features at the classification time. This assumption may not be valid in a real scenario where the data is often incomplete and segmented. Health care data suffers from a large volume of missing data due to the fact that not all the features are observed and collected (lab results, vital signals readings, etc) for all the patients at all time. One of the most common methods to fill out these missing values is to perform mean imputation. However, this practice has been shown to introduce more noise into the model rather than reduce it [17]. To tackle this problem, previous approaches segment the patient features according to their age group and then calculate the average value for each segment [18, 15]. Other methods handle missing values by fitting a distribution for each feature with the observed data and sample from the estimated distribution when the value is missing [14]. Similarly, the use of Multiple Imputation to predict the missing values has been proposed previously. Here regression techniques with the other observed features as covariates are deployed [18, 17]. Overall, these methods do not take into account the temporal aspect of the missing data where some features are highly dependent on previous values.

Most of the existing prediction models do not use text from the Electronic Medical Record (EMRs) due to its complexity. However, text data contains key information that is potentially useful to better predict the presence of an increase in the probability of mortality [10, 16]. Examples of text include lab reports, admission, doctors and nurse notes. Ghassemi et. al [10] combine static numerical features such as SAPS II score with topic modeling features from the text of the EMRs. They predict the probability of mortality inside the hospital and after the patient is discharged using Support Vector Machines (SVM).

We propose a dynamic method based on Bayesian Time Series to estimate the probability of mortality and to indicate the existence of a medical alarm. Our contribution is summarized as follows:

1. We model the probability of mortality as an aggregated latent state which is updated each time new features (lab results, vital signals, etc) are observed .
2. Our model is fed with heterogeneous source data from the EMRs (text or numerical data, and both discrete and continuous variables).
3. We incorporate the text information into the model by developing a method to convert the unstructured text information into discriminative features that are later incorporated into the model.
4. We address the missing values problem by estimating those values using a Regularized Expectation Maximization based method.

Table 1 shows a comparison of our proposed method with respect to other methods existing in the literature discussed above. We validate our model using Electronic Medical Records from patients admitted to the ICU. This data is obtained from the MIMIC II dataset [20]. By using a dynamic model, we predict the probability of mortality before the 24 hour window is complete. As a result, medical alarms can be triggered earlier as opposed to static methods.

This paper is organized as follows: in section 2 we explain how we construct our proposed framework and how we integrate each component into the model. Validation framework, experimental settings and empirical results are presented in section 3. Finally a discussion about the significance and impact of our proposed model is provided in section 4.

Table 1: Literature Comparison

Method	Dynamic Update	Features	Handles Missing Values
Apache III & SAPS II	No	Static Numerical	No
Batal 2009	No	Dynamic Numerical	No
Hug 2009	No	Dynamic Numerical	No
Lehman 2012	No	Static (Text + Numerical)	No
Gassemi et al 2014	No	Batch Aggregated Text + Static Numerical	No
Proposed Method	Yes	Dynamic (Text + Numerical)	Yes

2. METHODOLOGY

In this section, we describe the method to construct the probability of mortality as a latent state and the framework we use to handle missing values. In addition, we outline the methodology to process the text information to extract discriminant features.

2.1 Definition of Probability of Mortality as a Latent State

We define a patient i to be alive $Y_{t,i} = -1$ or dead $Y_{t,i} = 1$ at time t at the ICU as a binary variable. $Y_{t,i}$ has a Bernoulli distribution with probability of $\pi_{t,i}$, which we define as the probability that a patient i dies inside the ICU at time t (in hospital mortality). This probability is a function of a latent state $\theta_t = [\xi_{t-1,i}, \tilde{\theta}_{t,i}]'$ at time t. The value of $\tilde{\theta}_t$ is calculated by combining a set of observed features $X_{t,i}$ (measurements and procedures) and the value of the latent state at time $t-1$, $\tilde{\theta}_{t-1,i}$. The value of ξ_t reflects the log-odds effect on the probability of mortality $\pi_{t,i}$ by previous observed features contained in the state $\theta_{t,i}$.

In this framework, we are able to include both the features and health context from previous observations. This is not accounted for in the static classification frameworks. Our proposed model is a special case of the Generalized Dynamic Linear Models [24]. Here we employ the logit transformation to accommodate our specific context. This leads to the following expressions:

$$Y_{t,i} \sim Bernoulli(\pi_{t,i}), \tag{1}$$

$$\pi_{t,i} = \frac{e^{\xi_{t,i}}}{1 + e^{\xi_{t,i}}}, \tag{2}$$

$$\xi_{t,i} = \xi_{t-1,i} + \tilde{\theta}_{t,i} + w^{\xi}_{t,i}, \quad w^{\xi}_{t,i} \sim N(0, W_{\xi}) \tag{3}$$

$$\tilde{\theta}_{t,i} = \lambda \tilde{\theta}_{t-1,i} + \beta X_{t,i} + w^{\theta}_{t,i}, \quad w^{\theta}_{t,i} \sim N(0, W_{\theta}) \tag{4}$$

Here λ is a decay factor that determines the contribution of previous state values in the current one. The vector $X_{t,i}$ is constructed from the patient's observed lab test, vital signals, text, and demographics (features). In this model we

assume that most of the values of $X_{t,i}$ are observed. In the later subsections, we explain how we model and estimate the missing values in the feature vector. β represents the vector of regression coefficients we use to combine the observed features. W_ξ and W_θ are the evolution variances of ξ and $\tilde{\theta}$ respectively.

To illustrate the effect of previous outputs $\xi_{t,i}$ in the current state, we calculate the impact of the user's features $X_{t,i}$ observed at time t and then aggregate them into the state $\theta_{t,i}$ after k steps assuming no other value of $[X_{t+1,i} \dots X_{t+k,i}]$ is observed . This impact is determined by the following forecast function:

$$\xi_{t+k} = \sum_{r=0}^{k} \lambda^r \tilde{\theta}_t = \tilde{\theta}_t(1-\lambda^{k+1})/(1-\lambda) \tag{5}$$

As illustrated by the previous equation, the proposed model incorporates knowledge from prior measurements into the current state estimation. This effect representation allows us to predict patient probability of mortality even when no measurements are available at a given time t+k. In addition, the effect does not decrease over time, as opposed to the state evolution $\tilde{\theta}_t$. Each time there are new observations available, the value of the effect $\xi_{t,i}$ is updated using equation 3. The value of $\tilde{\theta}_{t,i}$ can take both positive and negative values. Thus, we are able to increase or decrease the probability of mortality using the observed features $X_{t,i}$

The model described above in equations 1-4 can be rewritten as a Dynamic Linear Model (DLM) as follows:

$$\begin{aligned} Y_{t,i} &\sim Bernoulli(\mu_{t,i}), & \mu_{t,i} &= \frac{e^{F'\theta_{t,i}}}{1+e^{F'\theta_{t,i}}}, \\ \xi_{t,i} &= F'\theta_{t,i} & \theta_{t,i} &= G\theta_{t-1,i} + [0, \beta X_{t,i}]' + w_t \\ F' &= [1,1] & \theta_{t-1,i} &= \begin{bmatrix} \xi_{t-1,i} \\ \tilde{\theta}_{t-1,i} \end{bmatrix} \\ G &= \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} & w_t &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} W_\xi & 0 \\ 0 & W_\theta \end{bmatrix}\right) \end{aligned} \tag{6}$$

Figure 1 shows the graphical model of this framework. The colored circles represent the variables that are observed in the model. The non-colored circles are the latent variables and model parameters that need to be inferred. The learning across multiple users is reflected through the estimated parameters Φ defined as:

$$\Phi = \lambda, W_\theta, W_\xi, \beta. \tag{7}$$

This representation is flexible enough to expand the model and to incorporate different weighting vectors β for different patient groups with a particular disease or age range.

The probability likelihood for observation $Y_{t,i}$, described in Equation 1-4, follows a Bernoulli distribution. Thus we transform probability of mortality $\pi_{t,i}$ to $\xi_{t,i}$ using the logit transformation. This model is similar to standard logistic regression as we use the same transformation. However, we incorporate the user features into an aggregated patient state that evolves over time, in contrast to static classification models. This approach allows us to predict future values of the state as more readings become available. Consequently, we are able to dynamically estimate the current patient probability of mortality and its evolution to predict an increase or decrease of this probability using the predictive forecast of the latent state. Algorithm 1 describes the fitting steps for the proposed model. The evolution of the state $\theta_{t,i}$, and the parameter estimation steps, are discussed in the following subsections.

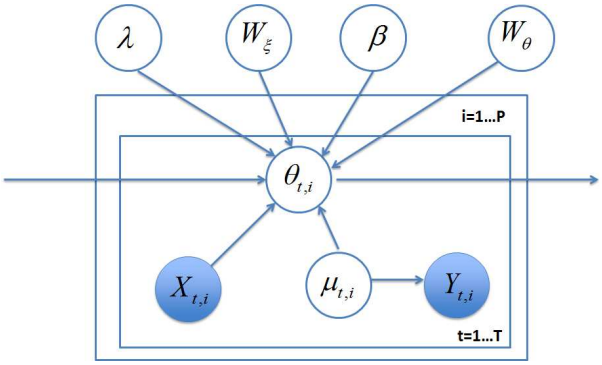

Figure 1: Graphical model of the patient health state

Algorithm 1 Proposed Method

```
Extract text and topic Features
Construct time series
Impute Missing Values using EM based method
Use an initial guess of the parameters Φ = (λ, W_ξ, W_θ, β)
Define P to be the number of patients in the dataset
repeat
  for i ← 1 to P do
    Define T_i to be the number of time steps of the series for
    a patient i
    for t ← 1 to T_i do
      Estimate the value of m_{t,i} and C_{t,i} using Filtering
      Equations described in subsection 2.2.1
    end for
    for t ← T_i to 1 do
      Estimate the expected value s_{t,i} = E(θ_{t,i}|D_{1:T_i}) and
      variance S_{t,i} = var(θ_{t,i}|D_{1:T_i}) of the hidden states us-
      ing FFBS algorithm described in subsection 2.2.2
    end for
  end for
  Estimate the parameters Φ that maximize the likelihood
  P(θ, Φ|D_{1:T_i,1:P})
until Convergence
```

2.2 Filtering Update and Model Inference

In this subsection, we develop the filtering update method to incorporate binary observations $Y_{t,i}$ sequentially to model the probability of mortality and its evolution. Then, we outline the backward smoothing recursion required to estimate the optimal model parameters. Finally, we describe the Expectation Maximization (EM) based method used to estimate the model parameters.

2.2.1 *Kalman Filtering Update*

We use the Kalman filtering equations to update the latent state using Dynamic Linear Models (DLMs) when series observations become available, $p(\theta_{t,i}|\Phi, D_{1:t,i})$ for $t = 1, \dots, T$ where $D_{1:t,i} = [X_{1:t,i}, Y_{1:t,i}]$ is the observed data up to time t and the model parameters Φ. Here, the observations are assumed to be continuous and normally distributed as defined in the standard DLM [24, 19]. However, our observations $Y_{t,i}$ are discrete. The main challenge in incorporating these observations into the model is the lack of a conjugate prior distribution for $\pi_{t,i}$, and consequently for $\theta_{t,i}$, which prevents us from estimating this probability in closed form.

Thus, we approximate the filtering update distribution by a normal distribution using the Laplace approximation [8].

Here, $\theta_{t,i}$ is approximated to a Normal distribution using the Maximum A Posteriori (MAP) estimate as the mean, and the Hessian of the log posterior distribution evaluated at the MAP as the variance. The log posterior distribution is approximated as:

$$\begin{aligned}\ell(\theta_{t,i}|D_{1:t,i},\Phi) \approx \quad & -\frac{1}{2}(\theta_{t,i}-a_{t,i})' R_{t,i}^{-1}(\theta_{t,i}-a_{t,i}) \\ & -ln(1+exp(-Y_{t,i}F'\theta_{t,i}))\end{aligned} \tag{8}$$

where $\theta_{t,i}|D_{1:t,i}$ is the state value for the patient i at time t given the complete observed data $D_{1:t,i}$ and the model parameters ϕ. Let $a_{t,i}$ be the predictive mean and $R_{t,i}$ be the predictive variance given $X_{t,i}$,$D_{1:t-1,i}$ for a patient i at the time t.Thus, we have:

$$\begin{aligned}a_{t,i} = \quad & Gm_{t-1,i} + \beta X_{t,i} \\ R_{t,i} = \quad & GC_{t,i}G' + W\end{aligned} \tag{9}$$

By letting $\theta_{t,i}^{MAP}$ and $H(\theta_{t,i}^{MAP})$ be the MAP state estimate and the Hessian respectively, filtering update process leads to following state expressions:

$$\begin{aligned}& & \theta_{t,i}|D_{1:t,i},\Phi \sim N(m_{t,i},C_{t,i}), & \\ m_{t,i} = \theta_{t,i}^{MAP}, & & C_{t,i} = -H^{-1}(\theta_{t,i}^{MAP}) & \\ \theta_{t,i}^{MAP} = & & \arg\max_{\theta_{t,i}} \ell(\theta_{t,i}|D_{1:t,i},\Phi) & \\ \frac{\partial \ell(\theta_{t,i})}{\partial \theta_{t,i}} = & & \frac{FY_{t,i}}{1+\exp\{Y_{t,i}F'\theta_{t,i}\}} - R_{t,i}^{-1}(\theta_{t,i}-a_{t,i}), & \\ \frac{\partial^2 \ell(\theta_{t,i})}{\partial \theta_{t,i}^2} = & & \frac{-FF'}{(1+\exp\{-Y_{t,i}F'\theta_{t,i}\})(1+\exp\{Y_{t,i}F\theta_{t,i}\})} & \\ & & -R_{t,i}^{-1} &\end{aligned} \tag{10}$$

Based on these derivatives, we find the MAP estimate using the Newton-Raphson iterative method since the closed form maximization is not feasible, leading to the following:

$$\begin{aligned}& \theta_{t,i}|D_{1:t,i} \sim N(m_{t,i},C_{t,i}), \qquad m_{t,i} = \theta_{t,i}^{MAP} \\ C_{t,i}^{-1} = \quad & \frac{FF'}{(1+\exp\{-Y_{t,i}F'\theta_{t,i}^{MAP}\})(1+\exp\{Y_{t,i}F\theta_{t,i}^{MAP}\})} \\ & +R_{t,i}^{-1}\end{aligned} \tag{11}$$

In this manner, we learn and update the state distribution from time $t-1$ to t. This estimation replaces the standard Kalman filtering equations to incorporate binary outputs and to model the latent patient probability of mortality.

2.2.2 Forward Filtering Backward Smoothing

We use Forward Filtering Backward Smoothing (FFBS) method to estimate the expected value of the states $\theta_{t,i}$ given the parameters in a DLM Φ [19, 9]. By using FFBS, we find the mean $m_{t,i}$ and $C_{t,i}$ variance of the hidden states distribution $p(\theta|D_{1:t,i}) \sim N(m_{t,i},C_{t,i})$ given the observations up to time t using the filtering equations described in subsection 2.2.1. Then, we obtain the smoothing mean $s_{T,i}$ and variance $S_{T,i}$ of the state variables at time T, $\theta_{T,i}|\theta_{D_{1:T_i,i}} \sim N(s_{T,i},S_{T,i})$. Conditional on this value, we estimate the mean $s_{T-1,i}$ and variance $S_{T-1,i}$ the state variables at time $T-1$ (backwards).

By combining these two steps, we guarantee the construction of a fully dynamic model with feedback. One variant of the model is a dynamic model with open loop feedback (no feedback about the future) by only fitting the model using forward filtering (FF) using the filtering equations of subsection 2.2.1. For numerical stability, we use the singular value decomposition (SVD) based approach detailed in [25] to find the values of $s_{t,i}$ and $S_{t,i}$. The steps needed to perform the FFBS in our model are described in Algorithm 2.

Algorithm 2 Forward Filtering Backward Smoothing

Estimate $p(\theta_{t,i}|\Phi,D_{1:t,i}) \sim N(m_{t,i},C_{t,i})$ for $t=1,\ldots,T_i$ as discussed in subsection 2.2.2.
Estimate $\theta_{T_i,i}|D_{1:T_i,i} \sim N(m_{T_i,i},C_{T_i,i})$

for $t \leftarrow T_i - 1$ to 1 **do**
Estimate $\theta_{t,i}|\theta_{t+1,i},D_{1:T_i,i} \sim N(s_{t,i},S_{t,i})$ (Backward Smoothing)

$$\begin{aligned}s_{t,i} &= m_{t,i} + C_{t,i}G'R_{t+1,i}^{-1}(s_{t+1,i}-a_{t+1,i}) \\ S_{t,i} &= C_{t,i} - C_{t,i}G'R_{t+1,i}^{-1}(R_{t+1,i}+S_{t+1})R_{t+1,i}^{-1}GC_{t,i} \\ R_{t+1,i} &= GC_{t,i}G' + W_{t,i} \\ a_{t+1,i} &= Gm_{t,i} + \beta X_{t+1,i}\end{aligned} \tag{12}$$

end for

2.2.3 EM based Parameter Maximization

For the parameter estimation, we use an Expectation Maximization (EM) method to estimate the value of the parameters. In the E-step, we estimate the expected state values using the Forward Filtering Backward Smoothing (FFBS) algorithm, explained in previous subsection. In the M-step, we estimate the parameters that maximizes the likelihood function estimated in the E-step.

The likelihood function is defined as follows:

$$p(\theta,\Phi|D_{1:T}) = \prod_{i=1}^{P} p(\theta_{1,i}) \prod_{t=2}^{T_i} p(\theta_{t,i}|\theta_{t-1,i}) p(\theta_{t,i}|\phi,D_{T_i}) \tag{13}$$

The log-likelihood is concave given G,λ, W_θ and W_ξ. As a consequence, the maximum likelihood estimate (MLE) is unique. We take the derivatives of the log-likelihood with respect to each of the model parameters and then setting them to 0. Based on the logic used in standard parameter estimation for Linear Dynamical Systems [9] and after some algebra, we obtain the M-step update expressions given the current value of parameters as follows:

$$\begin{aligned}G^{new} = & \left[\sum_{i=1}^{P}\sum_{t=1}^{T_i} E[\theta_{t-1,i}\theta_{t,i}']\right]' \left[\sum_{i=1}^{P}\sum_{t=1}^{T_i} E[\theta_{t-1,i}\theta_{t-1,i}']\right]^{-1} \\ E[\theta_{t-1,i}\theta_{t,i}'] = \quad & m_{t-1,i}s_{t,i}' + L_{t-1,i}\left(S_{t,i} + (s_{t,i}-a_{t,i})s_{t,i}'\right) \\ & -\beta X_{t,i}s_{t,i}' \\ E[\theta_{t-1,i}\theta_{t-1,i}'] = \quad & S_{t-1,i} + s_{t-1,i}s_{t-1,i}'\end{aligned} \tag{14}$$

$$\begin{aligned}W^{new} = & \frac{1}{\sum_{i=1}^{P} T_i}\sum_{i=1}^{P}\sum_{t=1}^{T_i} E[\theta_{t,i}\theta_{t,i}'] - E[(G\theta_{t-1,i}+\beta X_{t,i})\theta_{t,i}'] \\ E[\theta_{t,i}\theta_{t,i}'] = \quad & S_{t,i} + s_{t,i}s_{t,i}' \\ E[G\theta_{t-1,i}\theta_{t,i}'] = & G\left[m_{t-1,i}s_{t,i}' + L_{t-1,i}(S_{t,i} + (s_{t,i}-a_{t,i})s_{t,i}')\right] \\ E[\beta X_{t,i}\theta_{t,i}'] = \quad & \beta X_{t,i}s_{t,i}'\end{aligned} \tag{15}$$

$$\beta^{new} = \left[\sum_{i=1}^{P}\sum_{t=1}^{T_i}(s_{t,i}-Gs_{t-1,i})'X_{t,i}\right]\left[\sum_{i=1}^{P}\sum_{t=1}^{T_i} X_{t,i}'X_{t,i}\right]^{-1} \tag{16}$$

where:

$$\begin{aligned}& L_{t-1,i} = C_{t-1,i}G'R_{t,i}^{-1} \\ & p(\theta_{t,i}|D_{1:t-1}) \sim N(a_{t,i},R_{t,i}) \\ & R_{t+1,i} = GC_{t,i}G' + W_{t,i} \\ & a_{t+1,i} = Gm_{t,i} + \beta X_{t+1,i} \\ & p(\theta_{t,i}|D_{1:t,i}) \sim N(m_{t,i},C_{t,i}) \\ & p(\theta_{t,i}|D_{1:T,i}) \sim N(s_{t,i},S_{t,i})\end{aligned} \tag{17}$$

λ^{new}=$G_{2,2}^{new}$, W_{ξ}^{new}=$W_{1,1}^{new}$, W_{ξ}^{new}=$W_{2,2}^{new}$.

$m_{t,i}$ and $C_{t,i}$ for $t = 1 \dots T_i$ are calculated during the time series filtering step described in subsection 2.2.1. Given these estimations, backward smoothing is performed, based on standard Kalman smoothing recursion described in subsection 2.2.2, to obtain $s_{t,i}$, $S_{t,i}$. This process represents the E-step of the EM algorithm. The values of $a_{t,i}$ and $R_{t,i}$ represent the predictive expected state and the predictive state variance.

2.3 Missing Features Estimation

The proposed framework mentioned above assumes that all the patient's features are observed at each point in time. When feature values are not observed, we indicate they are missing and then we impute their value. For the current application, one portion of the missing data has a non-ignorable nature. For instance, the lack of observed values implies that the patient does not required certain lab or procedures to be performed. In addition, patient's features have an implicit temporal aspect. The value of features in time t are highly dependent on their value in previous time steps $1..t-1$. Then, standard imputation methods based on the mean value could lead to estimation errors.

To overcome these challenges, we impute the missing values by means of a Regularized Expectation Maximization method [22]. This method uses a regularization parameter to ensure the existence of positive definitive matrices needed to impute the missing data accurately.

The E-step consists of estimating the missing features $\hat{x}_m$ using the values of the available ones x_a inside the record using the following equation:

$$\hat{x}_m = \hat{\mu}_m + (x_a - \mu_a)\gamma, \quad \gamma \sim N(0, \Sigma) \tag{18}$$

where $\hat{\mu}_m$ is the mean estimate of the missing features of a record and μ_a is the mean estimate for the available features in a given record. We define as a record all the observed features of a patient at a given point in time $X_{t,i}$. The value of γ is the vector of regression coefficients for the available features. The intuition behind this step is to represent the missing record values as a combination of the available values for a given record and an estimated mean of the missing values of all the records.

After the missing values have been imputed, we perform the M-step. In this step, we estimate the values of the sample mean $\hat{\mu}$, the sample covariance matrix $\hat{\Sigma}$ and the coefficients $\hat{\gamma}$ using the ridge regression. The covariance matrix $\hat{\Sigma}$ is often negative definitive when the number of missing values is large. To overcome this issue, a regularization parameter is used to ensure the existence of a positive definitive matrix for $\hat{\Sigma}$. We run this method iteratively until convergence. More details of the algorithm can be found in [22].

2.4 Text Processing

Standard approaches to estimate the probability of mortality, Apache III and SAPS II scores, do not incorporate text information. One of the main challenges researchers face is to incorporate this type of data effectively.

We extract text features to improve the health state prediction. The text entries found in an EMR mainly consist of nurse's entries, procedures reports, admission and discharge information, among others. Each text entry has an assigned timestamp. Thus we are able to construct a time series for each of the text features we extract. In this subsection, we describe the steps we follow to process the text and extract different features that are later integrated into the model. Algorithm 3 shows a summary of the feature extraction process.Figure 2 depicts the text based feature representation.

Algorithm 3 Feature Extraction

```
Numerical Feature Extraction:
Transform features using Apache Score III weights
Perform χ² test on all the observed features
Select the features with higher separation score
Term based Feature Extraction:
Construct term frequency matrix
Estimate the χ² score for all terms and retain those with highest score
Classify each text entry as not improving (1) or improving (-1) using term features
Topic based Feature Extraction:
Fit topic Model
Estimate topic Mixture for every text entry
Determine presence or of each topic
Classify each text entry as not improving (1) or improving (-1) using topic features
```

2.4.1 Noun Phrases Extraction

Given the nature and domain of the text data, we need to extract meaningful phrases and concepts to obtain discriminative features and to improve our statistical estimates. To achieve this task, we extract noun phrases relevant to the medical domain that together with single terms are used in the text feature extraction process of daily text notes which are latter integrated in the dynamic model.

We extract relevant noun phrases by annotating the discharge summaries no included in the training data using the Clinical Text Analysis and Knowledge Extraction System (cTAKES)[21] and Metamap [1]. These Natural Language processing tools for the medical domain extract clinical named entities such as drugs, diseases/disorders, signs/symptoms, anatomical sites and procedures. We select the Discharge Summaries because these documents often aggregate the patient's medical history in a single document and provide us a richer set of noun phrases when compared with daily notes. This history includes all the patient's information collected during his/her stay in the ICU, together with past medical history and treatments and care after patient discharge.

After extracting annotating and extracting all the noun phrases, we select only those which describe a disease, a procedure or a medication using the medical ontologies provided by SNOMED [23]. In addition, we also detect which set of noun phrases corresponds to stop words (i.e. patient name, doctor name). Once we extracted the noun phrases, we note that some of the resulting phrases are a combination of two or more smaller medical noun phrases. Therefore, we decouple these phrases into their smallest possible unit. By means of tf-idf term selection, we select the most important noun phrases and remove those with low score.

Once the phrase selection is completed, we perform standard stop words removal and stemming before indexing the daily text entries using the extracted noun phrases and the single terms. Then, we extract two types of features: term and topic based features which are described below.

2.4.2 Term Based Features

We incorporate into the model a term-based feature using the obtained noun phrases and terms from the EMR text

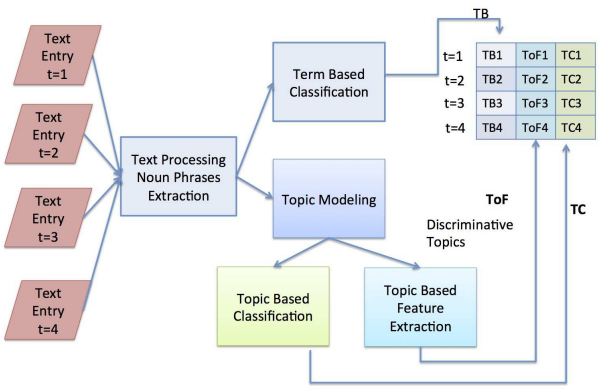

Figure 2: Text-based Feature Extraction Process

entry. This feature is the classification output of the text entries: -1 if the patient is recovering and 1 if not. Due to the fact that only a few number of people actually die while in the ICU, we have a set of unbalanced classes in our classification. We use the Naive Bayes classifier, which has been shown to provide good predictive performance and it is computationally feasible for this problem [7].

In order to make this classification feasible, we reduce the large vocabulary size of the corpus by extracting the most discriminative terms by means of χ^2 test [6]. We performed this test on the whole corpus. Our goal is to obtain a global estimate from the whole corpus in order to reduce the bias resulting from the term selection.

2.4.3 Topic Based Features

The second set of text features is based on statistical topic modeling. These models allow us to reduce the dimensionality of the term space to a smaller feature space of latent "topics". In addition, we are able to model topics for unseen documents without training the model again, as the method is generative.

In this context, each document is represented as a mixture of topics with a certain probability. Each topic is represented as a mixture of words. Our hypothesis is that topics can capture the global context of the document while this cannot be achieved by selecting text terms alone. By capturing this context, we are able to improve the performance of the probability of mortality estimation.

For this application, we fit a GD-LDA model developed by authors of [5] to extract the topics from the corpus set. This consists in all the processed text entry notes (noun phrases + terms) of all the patients. GDLDA, which is a generalization of LDA [3], allows us to model correlations between topics as opposed to LDA. In addition, this method is fitted in an unsupervised form, and it is computationally efficient, which permits us to train a large number of documents in a single batch contrary to other statistical topic models that model correlations such as Correlated Topic Models (CTM) [4].

We then remove the background topics which we define as word mixtures with a high percentage of common words (more than 90% of the terms inside the topic). We define as common words those that do not have healthcare related information by comparing them with the ontologies from the UMLS using MetaMap[1]. These ontologies provide information about healthcare treatments, drugs and diseases.

After removing the background topics, we select 10 most discriminative topics by means of the χ^2 test and include them in the dynamic model as features. In order to make the documents comparable, we use the values of $\{1, 0\}$ to show the presence or absence of a topic in the document instead of the probability of the topic in the document (two patients with similar medical history can have the same topics in their EMRs, but in different proportions). Thus, we indicate that a topic is present in a document if it accounts for more than 5% of the total topic mixture inside the document. In addition to the most discriminative topics, we include the classification output of the text entry (patient improving or not) using the document topic mixture as features. Here we use Naive Bayes classifier.

3. RESULTS

In this section, we depict the experimental settings to test and validate the proposed model. We also explain the feature extraction method for the numerical values. In addition, we report our experimental results using different performance measures. Finally, we discuss the impact of our proposed model in a real scenario.

3.1 Experimental Settings and Numerical Feature Extraction

We test our approach by predicting the mortality of a patient who is inside the Intensive Care Unit (ICU) and we trigger an alarm if this probability is larger than a certain threshold. It is critically important to assign medical resources effectively and to aid doctors and nurses ahead of time. For the situation which we are modeling, we predict the patient's mortality probability using the information available in his/her EMR. The EMRs are obtained from the MIMIC II data set [20]. This dataset contains text and numerical information that describes procedures, medications and vital signs readings from a given patient during his stay in the ICU. MIMIC II is composed of medical records from over 30,000 patients admitted to the ICU during a 7 year window. To validate our method, we use the medical records of $15,000$ patients selected randomly. In order to compare our approach with other methods from the literature, we only study the adult patients (over 18 years of age) without excluding patients due to an specific illness; this data consists of $11,648$ people.

A patient may be subject to different procedures and events during his stay in the ICU based on his condition. The events that are incorporated into the model are selected by means of the χ^2 test. We extract 30 features such as blood pressure level, lab procedures, pain level, and heart rate. We observe that the selected features are a combination of those used by the APACHE III and SAPS II scores [13, 12]. We find that 80% of the Apache III score features and 90% of the SAPS II features are included in our selected features. This selection shows consistency between these widely used method and our proposed approach.

We identify that some of the selected features are considered to have a bimodal distribution. For instance it is equally dangerous to have a really low blood pressure as to have it to be really high. To integrate this knowledge, we assign a weight for each possible range of the event. Those weights are obtained from the those used by the Apache Score III [13].

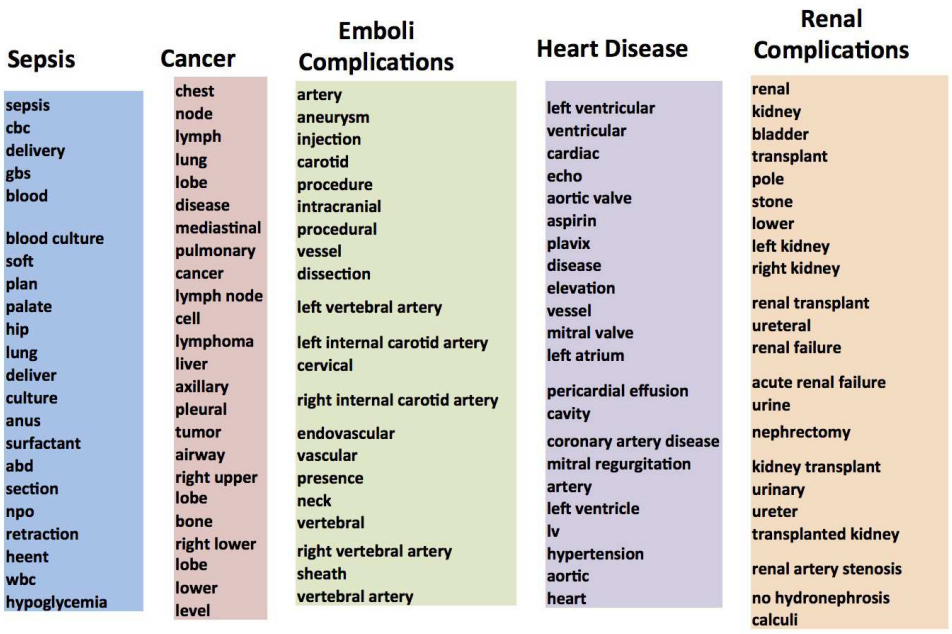

Figure 3: Sample of obtained topics from the text entries of MIMIC II dataset

We then divide the selected user features into two groups: static and dynamic. Some of the labs and procedures do not need to be performed at each time step. Therefore, we consider these type of features as quasi-static (they are updated if there is a new reading). Features such as blood pressure and pain levels are considered to be dynamic. This division impacts how missing values are treated in these features. The static features remain the same if they are not observed. Meanwhile, the dynamic features will be filled in using the Regularized Expectation Maximization method explained in subsection 2.3.

We construct a time series sequence for each selected feature events using the registered time stamp in three-hour increments. Due to privacy reasons, all time stamps are anonymized by adding a time offset to the entire patient series. However, our method requires time stamps relative to the patient admission to the ICU (t=0). Thus, we construct the relative series of events using the anonymized data and perform the analysis. We observe that approximately 57% of the patients stayed 24 hours or less. This means that common practices to predict the probability of mortality such as the Apache Score [13] and SAPS II [12] cannot be calculated for those patients.

In addition to the time of stay in the ICU, a large number of the patient observations are missing despite the time length of his/her stay. We estimate that 34% of the features used in the model are not observed during the patient's entire stay on average. This degree of missing values represents a challenge, because we need to infer the patient's health state (probability of mortality) even when no observations are available for these features. In addition to the temporal information, we include the patient gender and age as static features. These features allow us to establish the initial conditions of the patient's latent state.

To extract the text features, we consider each text entry to be a document. There are an average of 40 text entries per patient during his stay (a total of $582,592$ text entries). Each entry has an average length of 173 terms after constructing noun phrases, performing stemming, and removing stop words. We fit the GDLDA [5] model using all the text entries and $K = [50, 75, 100]$ topics. Figure 3 shows some of the obtained topics and how these topics are aligned with symptoms and procedures for a particular disease. Qualitatively, we observe that our topic modeling fitting (obtaining noun phrases, training the topic model with all the words in the document and filtering the resulting topics using the ontologies), provides topics which are cleaner and more interpretable than the ones provided by [10]. The resulting topics provide a better context which aids to improves the performance of the classification task. We determine that the best quality of topics is achieved on 75 topics. After estimating the topic mixture for each document, we remove the background topics as described above. We retain 65 topics after this step.

To validate our model, we select randomly 80% percent of the patients to be the training set and we use the remaining 20% as the test set. We report our results using a five-fold cross validation. We fit the model using three different set of features: numerical; numerical and term based; numerical, term based and topic modeling based.

We compare this information with a quasi-dynamic model we create, by predicting if a patient will die or not using static classification methods such as Random Forests and Naive Bayes. Here, we train a static classifier using the worst case scenario features at different times. In addition to these methods, we compare our method with three different score functions used in the literature: Apache III Score [13], SAPS II [12] and the results provided by Ghassemi et al [10].

3.2 Experimental Results

We fit the parameters of our proposed model using the Expectation Maximization approach described in section 2.2

Table 2: Performance of the 3 variations of our model, 3 different methods used in the literature and 2 static classification methods

	24 hours			48 hours			72 hours		
	Sensitivity	Specificity	AUC	Sensitivity	Specificity	AUC	Sensitivity	Specificity	AUC
Apache III	0.6925	0.1090	0.6769	-	-	-	-	-	-
SAPS II	0.6890	0.1393	0.6239	-	-	-	-	-	-
Random Forests	0.6208	0.1084	0.7421	0.6926	0.1004	0.7460	0.692	0.0910	0.7483
Ghasemi 2014[1]	0.638	0.850	0.8400	-	-	-	-	-	-
Naive Bayes	0.6559	0.1622	0.6367	0.5067	0.3243	0.6215	0.637	0.1916	0.6655
Proposed method Numerical Features	0.7038	0.5024	0.7606	0.7410	0.5564	0.6400	0.7310	0.62567	0.658
Proposed method Text and Numerical Features	0.7648	0.6924	0.7806	0.8071	0.4764	0.7445	0.7010	0.6567	0.6670
Proposed method Topic, Text and Numerical Features	0.7885	0.7905	0.8657	0.7822	0.7685	0.7985	0.7468	0.7992	0.73850

using 50 iterations for the 3 sets of features. We observe that the average estimated value of λ is around 0.02 for numerical features only and 0.21 for text and topic features. These values of λ imply that the current effect of the observations taken at time t is reduced to 10% after 4 and 3 steps respectively ($t+3$ and $t+4$). These step numbers are equal to the common delays between patient observations inside the ICU (9-12 hours).

We also evaluate the quality of the features obtained from the text. To achieve this, we calculate the precision from the estimated topic and text features for each text entry. We observe that when we use the topic based features at least one text entry of the people who died indicates that the patient is not recovering (around 70% of all the text entries were correctly classified); this signal is not evident when using term based features only. Here, people who die may not have any indication of worsening. This corroborates our hypothesis that topic based features result in better classification features than term based features.

We also show a comparison, between our proposed framework with the three variants we test, and static classification methods such as Random Forest with 50 trees and Naive Bayes with a sliding window, for 24, 48 and 72 hours. In addition, we compared our method with the Apache III and the SAPS II scores and the results reported by Ghassemi [10] which are calculated 24 hours after the patient is admitted to the ICU.

Table 2 shows the performance of our model in terms of sensitivity, specificity and AUC at 24, 48 and 72 hours after the patient enters to the ICU. In order to compare our results with methods in the literature, we report results in those time stamps, However our method is able to predict a probability of mortality every 3 hours. As we observe, our method which combines numerical, term based and topic based features has the highest AUC for 24 and 48 hours with respect to other methods, by at least 3.05% with respect to Ghassemi et al (0.8657 vs 0.8400) in 24 hours, and by 7.04% with respect Ramdom Forest (0.7985 vs 0.7460). We note that our models clearly outperforms all the other methods tested in specificity, (0.7905 of our method with the 3 different features vs 0.1622 of Naive Bayes at 24 hours).

Our method shows a better performance even when using only numerical features. This is due to the aggregation of dynamic features in previous times on the current latent state. Our method, with only numerical features, obtains better performance in terms of AUC than Apache Score (0.6925) and SAPS II (0.6899). In addition, our method is able to update the probability of mortality each time that a new observation is available; again this cannot be achieved with the scores mentioned above.

We estimate the sensitivity and specificity by selecting the highest sensitivity point of the ROC cuve. We observe that all three variants of our method have consistently better performance in terms of sensitivity and specificity than the other methods. Therefore, the use of text features and a dynamic model clearly improves the performance of mortality prediction significantly.

When comparing scores and static algorithms in the literature, we observe that Random Forests is the best static method in terms of performance. Intuitively, this method is the closest to what physicians do in the ICU to predict if a patient will survive after a period of time, based on their experience. Physicians tend to rule out diseases based on symptoms and vital signs values similar to a decision tree.

Note that the performance of our model decreases after 48 hours due to the variability among patients (from 0.8657 at 24 hours to 0.7985 at 48 for the numerical+term based+topics based in the AUC measure).

Table 3 shows the performance based on F-scores of our proposed model and other literature methods. We observe that our method outperforms other methods consistently for the different time intervals analyzed, 12.16% in 24 hours (0.59 for the 3 types of features vs 0.4662 of the Apache III Score), 23.69% in 48 hours (0.5450 with the 3 types of features vs 0.4406 of Ramdom Forests). Therefore, our method has a good predictive performance for the true cases.

In addition to the methods presented in previous tables, we test the performance of our method by removing the effect $\xi_{t-1,i}$ from equation 3 (reducing the model state from $\theta = [\xi_{t,i}, \tilde{\theta}_{t,i}]'$ to $\theta_{t,i} = [\tilde{\theta}_{t,i}]$). Here, we observe that the performance decreases dramatically (from an AUC of 0.82 to an AUC of 0.55 for the combination of the three types of features). This result confirms our hypothesis that the combination of previous values of the effect $\xi_{t,i}$ together with $\tilde{\theta}_{t,i}$ is more effective in estimating correctly the probability of mortality.

We also compare the performance of our method against Support Vector Machines (SVM). However, the results of this method are highly dependent on the imputation model used. Performing single value imputation with the missing

Table 3: Performance based F Scores of our model and related methods of the literature in $t = 24$, 48 and 72 hours after patient admission

Method	F Score		
	24 hours	48 hours	72 hours
Apache III	0.4662	-	-
SAPS II	0.3863	-	-
Naive Bayes	0.4137	0.4169	0.3975
Random Forest	0.3929	0.4406	0.3977
Proposed Method (only numerical features)	0.4629	0.5409	0.4570
Proposed Method (numerical + term based)	0.4929	0.5806	0.4853
Proposed Method (numerical + term + topic features)	0.5229	0.5450	0.5367

Table 4: Progression of the average probability of mortality in t=24,48,72 hours as predicted by our model with 3 different sets of features

Patients	24 Hours	48 Hours	72 Hours
Numerical Information			
Recovered	0.8321	0.8420	0.8420
Died	0.9500	0.9621	0.9620
Numerical Information + Text Features			
Recovered	0.8303	0.8395	0.8404
Died	0.9501	0.9695	0.9685
Numerical Information + Topic Features			
Recovered	0.8303	0.8321	0.8414
Died	0.9541	0.9685	0.9621

data leads to very poor results that are not comparable with the other methods shown in Table 2, thus we do not report the results of this technique.

Table 4 shows the progression of the probability of mortality of patients who died and patients that recovered and exit the ICU using the 3 variants of our model. Here we observe a clear difference in the probability of mortality between both groups of patients. Figure 3.2 shows the progression of the probability of mortality for a patient who died and a for a patient who recover and was discharged from the ICU and died few days later. Here, we observe that the patient who was discharged from the ICU increase its probability of mortality at the end of the series. Therefore, our framework is able to predict accurately this fact ahead of time even when the patient is discharged. Further work would include the estimation the probability of mortality after 30 days of discharge from the ICU. In addition, we plan to model the probability of reentry to the ICU, as this is another perspective measure of effective patient care.

3.3 Computational Complexity

Our model has a continuous state space compared to the discrete space of other models such as Hidden Markov Models (HMM). This implies that the quality of our prediction measure is superior to the discrete approximation of HMMs

[1]These are the results reported in [10]. Authors report results without cross validation. In addition authors remove patients with less than 100 stop words

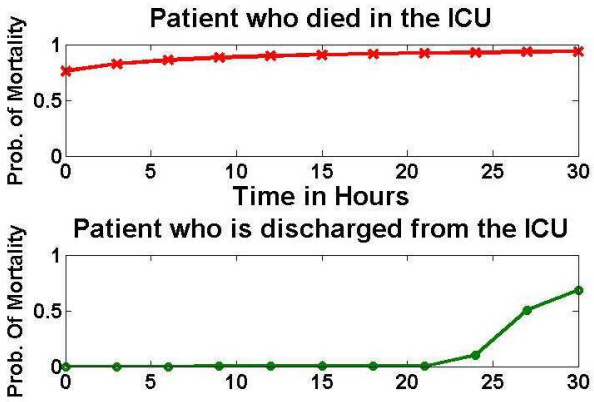

Figure 4: Progression of the probability of mortality for a patient who died (top), and for a patient who recovered and was discharged from the ICU (bottom) and died few days after being discharged

with less computational complexity. The complexity of the proposed approach is: $O(P * N^2 * T)$ where P is the of patients, N is the state dimension (2 for the current model), and T is the average series length for all the patients. On the other hand, the computational complexity of HMM based models would be exponential based on the number of adjacent states: $O(P * N^K * T)$ where N is number of prediction states (more than 4 if we want to model a minimum of granularity in the state transition), K is the number of adjacent states (likely to be large)) and T is the average time series size.

3.4 Impact of the proposed model in a Real Scenario

Our topic modeling fitting provides cleaner and more interpretable topics than the ones provided by the authors of [10].This extraction allows us to establish topics that are not only statistically meaningful but also semantically coherent according to a disease or treatment. This behavior is of vital importance to segment EMRs notes according to particular patient diseases.

Our method obtains significantly better specificity performance (percentage of true negatives detected) than other methods. A low specificity value implies the existence of a large number of false alarms. In a real scenario, this measure has a high impact due to the limited medical resources that health providers have. Physicians do not want to be overloaded with false alarms at the time a true alarm arrives. The proposed approach has higher F-score than other reported methods of the literature. This measure, which shows the ratio between the sensitivity and the positive predictive value, is very important in the correct detection of true alarms. Detecting all true alarms correctly is desirable since the cost of not detecting a patient who is very ill and dies is very high.

4. DISCUSSION, CONCLUSION AND FUTURE WORK

In this paper we have proposed a framework to exploit the dynamical information from the Electronic Medical Records of patients who are admitted to the ICU. Our method pro-

vides a fully dynamic framework that takes into account future uncertainty by training the model using the complete patient path from admission to discharge/death. Our model accounts for changes in the patient probability of mortality using dynamic features. We treat these features as stochastic processes and incorporate them into a latent state. This state modeling allows us to include a significant number of features with a moderate increase in complexity. In addition, we are able to capture and aggregate previous readings from the patient to estimate the current state, which cannot be achieved using static and quasi-dynamic models.

We demonstrate that the dynamic combination of text and numerical information improves the prediction performance. Text information gives a context that numerical features cannot provide. The use of these features have been shown to increase the performance of the mortality prediction [10]. Further extension of the work will include which other text features should be incorporated into the model to further improve the performance of the method. In addition, we plan to extract the main concepts and relationships between different patient's symptoms to better predict the probability of mortality.

Under the current model, we aggregate the patient information under a fixed schedule of data collection. We plan to include in our proposed framework to Adaptive Sampling methods to determine the optimal data sampling which can help us better predict the behavior of unseen patients. (Patients that are more critical should be sampled more frequently compared to patients that are more stable).

The method which we have developed opens the pathway to model each body subsystem (such as respiratory, digestive, cardiac) as an individual system that is later incorporated into a global estimate. This potentially could improve the performance in the prediction of patient's probability of mortality.

Acknowledgments

This work was partially supported by the NIST grant number 60NANB13D136, by NSF/NIST/UMBC grant number SC-0000015277, by CONACYT grant number 207751, by CITRIS SFP 2011-164, and by CITRIS SFP 2015-325.

5. REFERENCES

[1] A. R. Aronson and F.-M. Lang. An overview of metamap: historical perspective and recent advances. *JAMIA*, 17(3):229–236, 2010.

[2] I. Batal, L. Sacchi, R. Bellazzi, and M. Hauskrecht. A temporal abstraction framework for classifying clinical temporal data. *AMIA Annu Symp Proc*, 2009:29–33, 2009.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning*, 3:993–1022, 2003.

[4] D. M. Blei and J. D. Lafferty. Correlated topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. MIT Press, 2006.

[5] K. L. Caballero, J. Barajas, and R. Akella. The generalized dirichlet distribution in enhanced topic detection. In *CIKM*, pages 773–782, 2012.

[6] G. Forman, I. Guyon, and A. Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[7] E. Frank and R. R. Bouckaert. Naive bayes for text classification with unbalanced classes. In *In Proc 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 503–510, 2006.

[8] S. Geisser, J. Hodges, S. Press, and A. ZeUner. The validity of posterior expansions based on laplace's method. *Bayesian and likelihood methods in statistics and econometrics*, 7:473, 1990.

[9] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.

[10] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits. Unfolding physiological state: Mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 75–84, New York, NY, USA, 2014. ACM.

[11] C. W. Hug and P. Szolovits. Icu acuity: real-time models versus daily models. *AMIA Annu Symp Proc*, 2009:260–264, 2009.

[12] L. G. JR, L. S, and S. F. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Journal of American Medical Association*, 270(24):2957–2963, 1993.

[13] W. A. Knaus, D. P. Wagner, E. A. Draper, J. E. Zimmerman, M. Bergner, P. G. Bastos, C. A. Sirio, D. J. Murphy, T. Lotring, and A. Damiano. The apache iii prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults. *CHEST Journal*, 100(6):1619–1636, 1991.

[14] F. E. H. Kristel J.M. Janssena, A. Rogier T. Dondersb. Missing covariate data in medical research: To impute is better than to ignore. *Journal of Clinical Epidemiology*, (63):721–727, 2010.

[15] C. Lee, N. Arzeno, J. Ho, H. Vikalo, and J. Ghosh. An imputation-enhanced algorithm for icu mortality prediction. In *Computing in Cardiology (CinC), 2012*, pages 253 –256, sept. 2012.

[16] L.-w. Lehman, M. Saeed, W. Long, J. Lee, and R. Mark. Risk stratification of icu patients using topic models inferred from unstructured progress notes. *AMIA Annual Symposium Proceedings*, 2012:505–511, 2012.

[17] P. Liu, L. Lei, J. Yin, W. Zhang, W. Naijun, and E. El-Darzi. Healthcare data mining: Prediction inpatient length of stay. In *International IEEE Conference Intelligent Systems*, pages 832–837. Springer, 2006.

[18] A. Marshall, D. G. Altman, P. Royston, and R. L. Holder. Comparison of techniques for handling missing covariate data within prognostic modelling studies: a simulation study. *BMC Medical Research Methodology*, 10(7), 2010.

[19] G. Petris, S. Petrone, and P. Campagnoli. *Dynamic Linear Models with R*. use R! Springer-Verlag, 2009.

[20] M. Saeed, G. Lieu, and R. G. Mark. MIMIC II: a massive temporal icu patient database to support research in intelligent patient monitoring. *Computers in Cardiology*, 29:641–644, Sept. 2002.

[21] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, Sept. 2010.

[22] T. Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14:853–871, 2001.

[23] K. A. Spackman, P. D, K. E. Campbell, P. D, R. A. Cote, and D. S. (hon). SNOMED RT: A reference terminology for health care. In *J. of the American Medical Informatics Association*, pages 640–644, 1997.

[24] M. West and J. Harrison. *Bayesian forecasting and dynamic models (2nd ed.)*. Springer-Verlag, 1997.

[25] Y. Zhang and X. Li. Fixed-interval smoothing algorithm based on singular value decomposition. In *Proceedings of the Control Applications, 1996.*, pages 916 –921, sep 1996.