Title
Topic Models and Dynamic Prediction Models and their applications in Document Retrieval and Healthcare

Permalink
https://escholarship.org/uc/item/96m0z634

Author
Caballero Barajas, Karla Lizbeth

Publication Date
2015

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA
SANTA CRUZ

TOPIC MODELS AND DYNAMIC PREDICTION MODELS AND
THEIR APPLICATIONS IN DOCUMENT RETRIEVAL AND
HEALTHCARE

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Karla L. Caballero Barajas

September 2015

The Dissertation of Karla L. Caballero Barajas
is approved:

Professor Ram Akella, Chair

Professor Herbert Lee

Professor Donald M. Wiberg

Professor Ray R. Larson

Tyrus Miller
Vice Provost and Dean of Graduate Studies
# Table of Contents

List of Figures                                      vi
List of Tables                                       vii
Abstract                                             viii
Dedication                                           xi
Acknowledgments                                      xii

1 Introduction                                        1

I  Topic Models and Their Application in Document Retrieval  4

2 Generalized Dirichlet Topic Models                  5
  2.1 Introduction                                      5
  2.2 The Generalized Dirichlet distribution           7
    2.2.1 Properties and Intuition                     7
    2.2.2 Covariances: Properties and Constraints      10
  2.3 GD-LDA: Methodology                              11
    2.3.1 Notation                                     13
    2.3.2 Model and Gibbs Sampling                     13
    2.3.3 Parameter Estimation                         16
    2.3.4 Predictive Distributions                     17
  2.4 Experimental results                             19
    2.4.1 Validation                                   19
    2.4.2 Experimental Settings                        22
    2.4.3 Results                                      24
    2.4.4 Adhoc Document Retrieval                      31
  2.5 Discussion                                        35
3 Application of Topic Models in Relevance Feedback
3.1 Introduction ............................................. 37
3.2 Topic Models in Relevance Feedback .................... 39
3.3 Results .................................................. 41
3.4 Discussion .............................................. 43

4 Application of Topic Models in Healthcare related Document Retrieval
4.1 Introduction ................................................ 44
4.2 Statistical Topic Modeling ................................ 47
4.3 Noun Phrases Extraction .................................. 49
4.4 Feature Extraction from Discharge Summaries ............ 50
4.5 Retrieval Method ......................................... 51
4.6 Results .................................................. 51
   4.6.1 Experimental Settings ................................ 51
   4.6.2 Validation ........................................... 52
   4.6.3 Experimental Results ................................ 53
4.7 Discussion .............................................. 54

II Dynamic Models and Their Applications in Health State Prediction
5 Dynamic Health State Prediction .......................... 58
5.1 Introduction ............................................. 58
5.2 Dynamic Health State Prediction ......................... 62
5.3 Definition of Probability of Mortality as a Latent State .. 62
5.4 Filtering Update and Model Inference .................... 66
   5.4.1 Kalman Filtering Update ............................ 68
   5.4.2 Forward Filtering Backward Smoothing .............. 69
   5.4.3 EM based Parameter Maximization .................. 70
   5.4.4 Text Processing ..................................... 73
   5.4.5 Term Based Features ................................. 75
5.5 Missing Features Estimation ............................ 78
5.6 Estimation of Probability of Mortality ................... 80
   5.6.1 Experimental Settings and Numerical Feature Extraction .. 81
   5.6.2 Experimental Results ................................ 86
   5.6.3 Computational Complexity .......................... 90
   5.6.4 Impact of the proposed model in a Real Scenario ........ 91
5.7 Discussion .............................................. 92
6 Dynamic Estimation of the Probability of Patient Readmission to the ICU using Electronic Medical Records 95
6.1 Introduction ................................................. 95
6.2 Related Work .................................................. 97
6.3 Definition of Probability of Readmission as a Latent State .......... 99
6.4 Model Fitting .................................................. 101
6.5 Experimental Settings and Validation .................................. 103
6.6 Results ....................................................... 105
6.7 Discussion ..................................................... 109

7 Subsystem Failure Prediction ........................................ 111
7.1 Modeling the Probability of Subsystem Failure as an evolving Latent State111
7.1.1 Filtering Update and Model Inference .......................... 116
7.2 Estimation of the Probability of Mortality .............................. 122
7.3 Feature Extraction and Missing Values Estimation ...................... 122
7.3.1 Numerical Features Extraction .................................. 123
7.3.2 Text Features Extraction ...................................... 123
7.3.3 Missing Features Value Estimation ............................... 124
7.4 Results ....................................................... 125
7.4.1 Experimental Settings ...................................... 126
7.4.2 Experimental Results ....................................... 133
7.5 Discussion ..................................................... 136

8 Conclusion ..................................................... 139

Bibliography ................................................... 144

A Generalized Dirichlet Distribution .................................... 159
A.1 First and Second Moments of the GD distribution .................... 159
A.1.1 Covariance Properties of GD Distribution ...................... 160
A.2 Maximization of $\alpha$ and $\beta$ for GD ........................... 161
List of Figures

2.1  Tree structure of the GD distribution ............................... 8
2.2  Graphical Model of GD-LDA ........................................ 12
2.3  Qualitative Example of GD-LDA .................................... 25
2.4  Inferred Topics Correlation graph .................................. 26
2.5  Qualitative Comparison Topic Models ............................... 27
2.6  Distribution of the number of significant topics per document .... 28
2.7  Mean per-document log-Likelihood ................................. 30
2.8  Average running times in minutes ................................. 33
2.9  Cumulative distribution of the topic mixtures .................... 34

5.1  Graphical model of the patient health state ....................... 66
5.2  Text-based Feature Extraction Process ............................ 76
5.3  Sample of obtained topics from the text entries of MIMIC II dataset ... 78
5.4  Progression of the probability of mortality ........................ 90

6.1  Fitting Steps for the proposed model .............................. 102

7.1  Graphical model Subsystem Failure ............................... 115
7.2  Graphical Model Combination Subsystem .......................... 115
## List of Tables

2.1 Features of the datasets analyzed. Mean document includes the 95% interval .................................................. 23  
2.2 Number of Gibbs samples per EM iteration .......................... 23  
2.3 Results of ad hoc IR using Topic Models ............................ 31  
3.1 Results of Topic feedback using 50 topics in the OHSUMED dataset .... 42  
4.1 Document features: Single Words and with noun Phrases extracted .... 52  
4.2 Mean Performance Results of the base model and the variants of the model for the test set .................................. 55  
5.1 Literature Comparison .................................................. 61  
5.2 Model Performance Probability of Mortality ................................ 83  
5.3 F Score Performance Probability of Mortality .......................... 89  
5.4 Progression of the average probability of mortality ......................... 89  
6.1 Performance of the 3 variations of our model, 2 different methods used in the literature and 2 static classification methods in terms of Sensitivity, Specificity and AUC .............................................. 108  
7.1 Subsystem Model Performance ......................................... 129  
7.2 Mortality Prediction using Subsystem Failure Prediction ................. 132
Abstract

Topic Models and Dynamic Prediction Models and their applications in Document Retrieval and Healthcare

by

Karla L. Caballero Barajas

Statistical Topic Models has been widely studied in Text Mining as an effective approach to extract latent topics from unstructured text documents. We present a robust and computationally efficient Hierarchical Bayesian model for effective topic correlation modeling Generalized Dirichlet distribution (GD). GD-LDA is effective to avoid over-fitting as the number of topics is increased. We provide results using Empirical Likelihood (EL) in 4 public datasets. We show the application of topic models in two different domains: 1)Information Retrieval, and 2)Dynamic Prediction Models applied in health care.

In Information Retrieval, we propose to leverage statistical topic modeling techniques in relevance feedback to incorporate a better estimate of context by including corpus level information about the document. We show results using the OHSUMED dataset for three different variants and obtain higher performance, up to 12.5% in Mean Average Precision (MAP).

Patients often search for information on the web about treatments and diseases after they are discharged from the hospital. However, searching for medical information on the web poses challenges due to related terms and synonyms for the same disease
and treatment. We present a method to retrieve healthcare related documents using the patient discharge document. We show that the proposed framework outperformed the winner of the retrieval CLEF eHealth 2013 Challenge by 68% in the MAP measure, and by 13% in NDCG.

We present a method to estimate dynamically 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. We update this state each time new patient data is observed. We test our proposed approach using 15,000 Electronic Medical Records (EMRs) obtained from the MIMIC II public data set.

We expand this dynamic mortality estimation model in two forms. We estimate the probability that a patient is readmitted after he is discharged from the ICU and transferred to a lower level care unit. We also present a method to predict the failure of physiological subsystems from patients admitted to the ICU using heterogeneous data dynamically. We model the probability of failure in each subsystem as a latent state. Then, we estimate the probability of patient mortality as a combination of the estimated failure propensity for all subsystems. We propose a method of imputing missing values using the non-ignorable nature of the patient data. Experimental results show that our method outperforms other approaches in the literature in terms of AUC, sensitivity, and specificity. In addition, we show that the combination of different features (numerical

ix
and text) increases the prediction performance of the proposed approach.
To my husband and children,

Joel, Dannah, Minerva and Cesar
I would like to thank and acknowledge all the support from many people and institutions who have contributed one way or the other to the completion of this dissertation.

First, I would like to thank my advisor Ram Akella for the opportunity to collaborate with different projects and companies from the Silicon Valley. Under his supervision, I had a chance to lead collaborations and propose solutions. The various interactions with industry and academia have helped me to form my professional skills and to be an independent thinker.

I would like to thank the different companies where I could participate in research projects, especially to SAP, and HP. Working with them provided me the opportunity to learn and work on real-life projects. A big thanks to Paul Hoffman and Manny Aparicio.

Also, I want to thank my thesis committee members, Prof. Herbie Lee, Prof. Donald Wilberg and Prof. Ray Larson for your flexibility for the defense and your feedback. Your insightful comments help me to review and improve this dissertation work.

I want to recognize and thank staff members and graduate advisors Emily Gregg, Kris West, Carol Mullane, Connie Miller, and Gisela Richards for helping me above and beyond the call of duty. They help me enormously in navigating the administrative system and provide me insightful advice on this long journey.

I wish to thank the following agencies and institutions for their continuing
financial support: CONACYT, UC Santa Cruz Regents, SAP, HP, and CITRIS. My graduate school experience would not have been possible without their support.

I want to thank my close and extended family for their unconditional support. To my mother who teach me that I can achieve whatever I want if I work towards it. She has been my role model as a working mom and as an excellent professional. Thank you, mom. A big thanks to my compadres Joaquin Salas and Maria Elena Meza whose advice in the personal and professional field have been precious to me.

I am infinitely thankful to my three children: Dannah, and my twins Cesar, and Minerva. They have been my teachers in life. They taught me how to combine personal and professional careers. I would like to thank my oldest daughter, Dannah, for the countless days she was willing to stay at the office so I could work on the dissertation. For being so understanding about deadlines and travels, and for her support and patience about the crazy schedule that implies to be a Ph. D student and mom at the same time.

Finally, I would like to thank my husband, Joel, who has been my professional colleague in this Ph. D journey and my personal teammate in life. Joel, you have been an incredible partner, husband, and father. I would like to thank you for your support in the best and worst moments. I can not wait for next chapter in our lives.
Chapter 1

Introduction

Statistical Topic modeling has been widely studied in Machine Learning and Text Mining as an effective approach to exploiting term co-occurrences in documents and to extracting latent topics from unstructured text documents. However, new approaches, which handle correlations between topics, are rarely used in large-scale datasets. These models have a high computational cost and low performance when handling a broad range of document numbers, vocabulary size and an average word length per document. Some of the applications of Statistical Topic Models include Document Retrieval and Feature Extraction and Reduction.

In document retrieval, Topic Models provide global information about documents. These methods provide an unsupervised clustering framework that enables us to expand the query and to increase the weight of related words without including them explicitly in the document or the query. In Relevance Feedback, Topic Models have showed to add global features and create clusters to improve the retrieval perfor-
mance significantly. Thus, these methods provide a robust and automatic method to incorporate context to the user feedback.

Statistical Topic Modeling provide us a form of Features Extraction that allows us to include a reduce set of text features into prediction models. They enable us convert the unstructured text information into discriminative features that are later incorporated into the prediction model.

In this thesis, we develop an efficient computational framework to model topic models while taking into account the correlation between topics. In addition, we present five different applications of Topic models as a form of feature extraction: 2 in the document retrieval domain and three in the dynamic prediction using heterogeneous data.

In the retrieval domain, we explore and develop a method to incorporate context using topic models inside the relevance feedback framework. Here, we incorporate the extracted topic features from positive and negative document labels as a form to improve retrieval performance. The second application is to include topic models as a form of smoothing in the retrieval of healthcare related documents. In this context, we include the topics from related documents such as the discharge summaries. We add context to the user’s query that mainly consists of diseases and treatments with acronyms.

In the dynamic prediction context, we present three applications that are related to the prediction of a patient’s health state as a decision support system to the physicians. This state is reflected in the probability of mortality and the probability
of patient’s readmission to the ICU, and the prediction of patient’s subsystem failure.

We develop methods that aggregate heterogeneous data into a latent state that evolves every time there is newly observed features using Dynamic Linear Models. The patient features are represented by a collection of heterogeneous data obtained from the Electronic Medical Records. These documents consist of a collection of vital signs readings; labs performed to the patient while he is in the ICU and doctor and nurse notes.

This thesis is organized as follows: In the first part we describe a methodology to extract Topic Models while capturing correlation in an efficient manner. In addition, we present two applications using Topic Models in Document Retrieval. The first application is using the topic mixture instead of words to perform Relevance Feedback. The second application is based on extracting topic mixtures from the discharge summaries. These are later used as a smoothing component to add context to the query and to include implicitly related words into the query.

In the second part of the thesis, we show a method to use dynamic models in the dynamic prediction of the patient’s health state. Here we use heterogeneous data such as text and numeric features obtained from the Electronic Medical Records (EMR) as an input. In this context, we show two different applications: prediction of patient mortality and estimation of the probability of readmission in the ICU. Then, we expand the model to accommodate the dynamic estimation of the probability of failure in different physiological subsystems. Those estimations are later combined to predict the probability of mortality inside the ICU. Finally, we present a global conclusion and future lines of research.
Part I

Topic Models and Their Application in Document Retrieval
Chapter 2

Generalized Dirichlet Topic Models

2.1 Introduction

Topic modeling has been widely studied in Machine Learning and Text Mining as an effective approach to extract latent topics from unstructured text documents. The fundamental idea underlying topic modeling is to use term co-occurrences in documents to discover associations between those terms. The development of Latent Dirichlet Allocation (LDA) [11, 35] enabled the rigorous prediction of new documents first time. Consequently, variants and extensions of LDA have been an active area of research in topic modeling. This research has 2 main streams in document representation: 1) The exploration of super and subtopics as in Pachinko Model Allocation (PAM) [54, 55]; and 2) the correlation of topics [12, 54, 55]. However, despite the improvement of these approaches over LDA, they are rarely used in applications that handle large data sets, such as Information Retrieval. Major gaps in these models include: modeling correlated
topics in a computationally effective manner; and a robust approach to ensuring good performance for a broad range of document numbers, vocabulary size and average word length per document.

In this chapter, we develop a new model to meet these needs. We use the Generalized Dirichlet (GD) distribution as a prior distribution of the document topic mixtures, leading to GD-LDA. We show that the Dirichlet distribution is a particular case of GD. As a result, GD-LDA is deemed to be a generalized case of LDA. Our goal is to provide a more flexible model for topics while retaining the conjugacy properties, which are desirable in inference. The features of the GD-LDA model include: 1) An efficient method to represent sparse topic correlations in natural language documents. 2) A model that handles global topic correlations with time complexity $O(KW)$, adding minimal computational cost respect to LDA. This results in a fast and robust approach compared to CTM and PAM. 3) A hierarchical tree structure that accommodates the most significant topics, based on probability mass, at the upper levels allowing us to reduce the number of topics efficiently. Note that GD is a particular case of Dirichlet Trees [68, 22]. This distribution has been used previously in topic modeling to add domain knowledge to the probability of words given a topic [4]. In contrast, we use the GD distribution to model topic correlations. Thus, this approach is complementary to GD-LDA.

To validate our model, we use Empirical Likelihood (EL) in four data sets with different characteristics of document length, vocabulary size, and a total number of documents. GD-LDA outperforms CTM, PAM, and LDA consistently in all the
data sets. Also, we test the performance of GD-LDA in ad-hoc Information Retrieval, obtaining superior results to those in the literature. We show a significant difference in running times between GD-LDA, CTM, and PAM, which makes GD-LDA as viable as LDA for large data sets.

Then, we present the performance of GD-LDA in ad hoc information retrieval (IR) based on MAP, P@10, and Discounted Gain. For all the IR measures, GD-LDA shows higher performance than LDA, the dominant topic model in IR. All these improvements with a small increase in fitting time than LDA, as opposed to CTM and PAM. Finally, we present a method to use the topic mixture.

This chapter is organized as follows: In section 2.2, we present the definition and features of the GD distribution. We will use this definition to develop the methodology of the GD-LDA model. Section 2.3 depicts our proposed approach with the proper derivations. Validation criteria, experimental settings, and results are presented in section 2.4. Finally, we present a discussion and future paths of research in section 2.5.

2.2 The Generalized Dirichlet distribution

2.2.1 Properties and Intuition

The Generalized Dirichlet (GD) distribution was introduced by Connor and Mosimann in [20]. The model of the GD distribution is motivated by the limitations of the Dirichlet distribution in modeling covariances. In the case of the Dirichlet distri-
distribution, all the entries of the random vector must share a common variance, and they
must sum to one. When we use the Dirichlet distribution as a prior for the multinomial
distribution, we have only one degree of freedom, the total prior sample size, to
incorporate our confidence in the prior knowledge. As a consequence, we cannot add
individual variance information for each entry of the random vector.

In addition, all entries are always negatively correlated. In other words, if the
probability of one entry increases each of the other probabilities must either decrease
or remain the same to sum to one. Despite these limitations, the Dirichlet distribution
is widely used given that this is a conjugate prior of the multinomial distribution.

The GD distribution allows us to sample each entry of the random vector of
proportions from independent Beta distributions. This independence is the key property
that provides more flexibility than the Dirichlet distribution. Formally this distribution
is defined as:

$$p(\theta|\alpha, \beta) = \prod_{j=1}^{K-1} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \theta_j^{\alpha_j-1}(1-\theta_1-\cdots-\theta_j)^{\eta_j}$$

(2.1)

where $\theta_1 + \theta_2 + \cdots + \theta_{K-1} + \theta_K = 1$, $\eta_j = \beta_j - \alpha_j+1 - \beta_{j+1}$ for $1 \leq j \leq K - 2$ and
$\eta_{K-1} = \beta_{K-1} - 1$.

To illustrate the properties of the GD distribution we define $Z_1 = \theta_1$ and
\[ Z_k = \frac{\theta_k}{V_k} \text{ for } k = 2, 3, \ldots, K - 1 \text{ where } V_k = 1 - \theta_1 - \cdots - \theta_{k-1}. \]

Let \( T_k \) be the discrete random variable with multinomial distribution with parameter \( \theta_1 \ldots \theta_K \) for \( K \) different categories. We start at node \( V_1 \), and at this node we sample \( T_1 \) with probability \( Z_1 \) and \( V_2 \) with probability \( 1 - Z_1 \). Conditional on \( V_2 \), we sample \( T_2 \) with probability \( Z_2 \) and \( V_3 \) with probability \( 1 - Z_2 \). In the general case, conditional on \( V_k \), we sample \( T_k \) with probability \( Z_k \) and \( V_{k+1} \) with probability \( 1 - Z_k \) for \( k = 1 \ldots K - 1 \). If we now add a prior Beta distribution with parameters \( \alpha_k, \beta_k \) for each conditional Binomial distribution of the nodes \( V_k \), we have a GD distribution where this set of Beta distributions is conjugate to the set of Binomial distributions.

\[ Z_k \sim Beta(\alpha_k, \beta_k) \quad T_k \sim Bin(Z_k, N_k) \text{ for } N_k = N - \sum_{i<k} T_i \quad (2.2) \]

where \( N \) is the total number of observations, and \( N_k \) is the number of observations remaining from previous categories in the tree[93].

The GD distribution is a special case of the Dirichlet Tree distribution [68, 22] where a cascade hierarchy is employed in the generative process of the distribution. To interpret the parameters of the GD distribution we refer to its tree representation is shown in Fig 2.1. Conditional on \( V_k \), \( \alpha_k \) is the sample size assigned to the discrete output \( T_k \), and \( \beta_k \) is the sample size allocated to the next level in the tree. If \( \beta_k \) is too small compared to \( \alpha_k \), we could discard the rest of the tree. This cascade allocation is a desirable property as it facilitates dimensionality reduction in the number of topics. By setting \( \beta_k = \alpha_{k+1} + \beta_{k+1}, \beta_{K-1} = \alpha_K \), we obtain a Dirichlet distribution. Therefore, the GD distribution ”generalizes” the Dirichlet distribution [20]. Although a general
Dirichlet Tree structure that is inferred by data is highly appealing, in practice this structure determines the number and properties of the parameters to fit. Thus, the tree must be fixed before inferring the model [68, 70]. A key property of the GD cascade structure is that it facilitates topic reduction because the tree can be pruned based on the conditional probability of going to the following level.

The GD distribution is a conjugate prior distribution to the Multinomial distribution, and then we can integrate out the parameter of the Multinomial distribution leading to:

\[
\begin{align*}
p(T|\alpha, \beta) &= \int p(T|\theta)p(\theta)d\theta \\
&= \prod_{k=1}^{K-1} \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \prod_{k=1}^{K-1} \frac{\Gamma(\alpha'_k + \beta'_k)}{\Gamma(\alpha'_k + \beta'_k)}
\end{align*}
\]

where \(\alpha'_k = \alpha_k + T_k\), \(\beta'_k = \beta_k + T_{k+1} + \ldots + T_K\). The ability to estimate this integral in closed form is crucial for GD-LDA model because this enables us to perform Gibbs sampling. This is the product of Beta-Binomial distributions for \(Z_k\) independent random variables. As a result, this expression is factorized into independent ratios of Gamma functions, which allows us to find the Maximum Likelihood Estimation (MLE) of \(\alpha, \beta\) in a simple manner (independently) which is not the case of the Dirichlet distribution.

### 2.2.2 Covariances: Properties and Constraints

In natural language documents, most of the times only a few topics co-occur leading to sparse topic correlations [90]. This fact implies that a very few random topics may suffice, rather than the full joint distribution including all the topics.

When considering useful distributions for topic modeling, a desirable feature is
to model only a few topics in a document. For instance, the topic domestic politics could frequently co-occur with middle East politics and health care reform. This implies that if we observe domestic politics, the conditional probabilities of observing middle East politics and health care reform will increase. However, if a document contains domestic politics and middle East politics, it could very rarely contain health care reform.

GD has the computational advantage of only having $2K - 1$ parameters. This reduced number of parameters also implies that the distribution covariances are constrained. The question is whether it is possible that these constraints imply that only a few topics co-occur in a document. In this respect, the covariance properties described in Appendix A.1, particularly the third one, indicate that the covariances are dependent on the expected value of two topics. Then several covariances might be close to zero if the right-hand side ratio is small. This dependency is an important feature of the GD distribution which we propose to exploit in topic modeling. A detailed discussion of the covariance constraints of GD distribution is provided in Appendix A.

### 2.3 GD-LDA: Methodology

In this section, we depict the parameter estimation process to fit the GD as a prior distribution for topics. We show that GD-LDA can be fitted with a computational cost similar to that of LDA. Fig 2.2 displays the graphical model for GD-LDA, and we describe its generative model in Algorithm 1.

We follow a Monte Carlo Expectation Maximization (MCEM) approach to fit
**Algorithm 1 GD-LDA Generative Model**

for topic $k \leftarrow 1$ to $K$ do

    draw $\phi_k \sim \text{Dir}(\gamma)$

end for

for document $j \leftarrow 1$ to $D$ do

    draw $\theta_j \sim \text{GenDir}(\alpha, \beta)$

    for word $w \leftarrow 1$ to $N_j$ do

        draw $z_{wj} \sim \text{Mult}(\theta_j)$

        draw $w|z_{wj} = k \sim \text{Mult}(\phi_k)$

    end for

end for

Figure 2.2: Graphical Model of GD-LDA. $\alpha$ and $\beta$ are vectors of size $K - 1$ where $K$ is the number of topics. $\gamma$ is a vector of size $V$ (vocabulary size).

Our model. Conditional on the hyper-parameters, we develop a Gibbs sampling approach to infer the topic assignments to each word in the corpus. Then, assuming that the topic assignment expectations as given, we optimize the hyper-parameters of the model.
2.3.1 Notation

Let $K$ be the number of topics, $D$ the number of documents, and $V$ the vocabulary size. We use the indices: $i$ to denote a word or term index in the vocabulary. We use $k$ to indicate a specific topic, $j$ to refer to a document, and $w$ to identify a specific observed word. $N_{i,k}$ defines the number of observed words that correspond to term $i$ and have been assigned to topic $k$. $N_{j,k}$ is the frequency of topic $k$ in document $j$. We refer to $p(w,z|·)$ as the joint probability of all the words $w$ in the corpus and their topic assignments $z$. $\gamma$ represents a vector of size $V$. $\alpha$ and $\beta$ are vectors of size $K - 1$.

2.3.2 Model and Gibbs Sampling

We define the joint probability associated with the graphical model defined in Fig 2.2 with joint distribution:

$$p(w, z|\alpha, \beta, \gamma) = p(w|z, \gamma)p(z|\alpha, \beta)$$  \hspace{1cm} (2.4)

This expression allows us to decompose the problem into two hierarchical models that can be treated and optimized separately based on these conditional probabilities. The probability of words given topics is:

$$p(w|z, \gamma) = \int_\phi p(w|\phi, z)p(\phi|z)\,d\phi = \prod_{k=1}^K \frac{\Gamma(\sum_{i=1}^V \gamma_i)}{\prod_{i=1}^V \Gamma(\gamma_i) \Gamma(\sum_{i=1}^V (\gamma_i + N_{k,i}))}$$ \hspace{1cm} (2.5)

For the probability of topics, we have a GD prior distribution for the topic mixtures in each document, which is assumed to be Multinomial. Thus we have:

$$p(z|\alpha, \beta) = \int_\theta p(z|\theta)p(\theta|\alpha, \beta)\,d\theta = \prod_{j=1}^D \prod_{k=1}^{K-1} \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \prod_{k=1}^{K-1} \frac{\Gamma(\alpha_j^k + \beta_j^k)}{\Gamma(\alpha_j^k + \beta_j^k)}$$ \hspace{1cm} (2.6)
where \( \alpha^j_k = \alpha_k + N_{j,k} \), \( \beta^j_k = \beta_k + N_{j,k+1} + \ldots + N_{j,K} \).

The topic assignments \( z \) are not observed. Then, we define a Gibbs sampling method to infer \( z \) as follows:

\[
p(z_w = k | z^{w_j}, \alpha, \beta, \gamma) = \frac{p(w | z, \gamma)p(z | \alpha, \beta)}{p(w | z^{w_j}, \gamma)p(z^{w_j} | \alpha, \beta)}
\]

Here, \( z^{w_j} \) represents the topic assignments for all the words except the word \( w \) from document \( j \). This analysis leads us to the following distributions for Gibbs sampling:

\[
\begin{align*}
\frac{p(w | z, \gamma)}{p(w | z^{w_j}, \gamma)} & \propto \frac{N_{i,k}^{w_j} + \gamma_i}{\sum_{i=1}^{V} (N_{i,k}^{w_j} + \gamma_i)} \\
p(z | \alpha, \beta) & \propto \begin{cases} \\
\frac{\alpha_k + N_{j,k}^{w_j}}{\alpha_k + \beta_k + \sum_{l=1}^{K} N_{j,l}^{w_j}}, & k = 1 \\
\frac{\alpha_k + N_{j,k}^{w_j}}{\alpha_k + \beta_k + \sum_{l=1}^{K} N_{j,l}^{w_j}} \prod_{m=1}^{k-1} \frac{\beta_m + \sum_{l=m+1}^{K} N_{j,l}^{w_j}}{\alpha_m + \beta_m + \sum_{l=m}^{K} N_{j,l}^{w_j}}, & 1 < k < K \\
\prod_{m=1}^{K-1} \frac{\beta_m + \sum_{l=m+1}^{K} N_{j,l}^{w_j}}{\alpha_m + \beta_m + \sum_{l=m}^{K} N_{j,l}^{w_j}}, & k = K
\end{cases}
\end{align*}
\]

In LDA, the topic distribution depends on \( \alpha_k \) alone for the current topic \( k \).

Intuitively, if a new sample is assigned to a topic \( k \) in LDA, there is no effect on the sampling distribution of other topics. However, if a new sample is assigned to a topic \( k \) in GD-LDA, it will affect other topics through the evaluation of the product in Eq 2.9. The impact of this assignment will depend on the parameters \( \alpha, \beta \).

Sampling from the distribution in Eq 2.9 results in a high computational cost due to the cumulative product defined there. Since this distribution is not standard,
Algorithm 2 GD-LDA Gibbs Sampling

\begin{align*}
\text{cumFactor} & \leftarrow 1, \ k \leftarrow 1 \\
\text{CDF}[k] & \leftarrow \frac{\alpha_k + N_{j,k}^{-w_j}}{\alpha_k + \beta_k + \sum_{l=1}^{K} N_{j,l}^{-w_j}} \times \frac{N_{i,k}^{-w_j} + \gamma_i}{\sum_{i=1}^{V} (N_{i,k}^{-w_j} + \gamma_i)} \\
\textbf{for} \ k \leftarrow 2 \ \text{to} \ K-1 \ \textbf{do} \\
\text{cumFactor} & \leftarrow \text{cumFactor} \times \frac{\beta_{k-1} + \sum_{l=k}^{K} N_{j,l}^{-w_j}}{\alpha_{k-1} + \beta_{k-1} + \sum_{l=k-1}^{K} N_{j,l}^{-w_j}} \\
\text{CDF}[k] & \leftarrow \text{CDF}[k-1] + \text{cumFactor} \times \frac{\alpha_k + N_{j,k}^{-w_j}}{\alpha_k + \beta_k + \sum_{l=1}^{K} N_{j,l}^{-w_j}} \times \frac{N_{i,k}^{-w_j} + \gamma_i}{\sum_{i=1}^{V} (N_{i,k}^{-w_j} + \gamma_i)} \\
\textbf{end for} \\
\text{CDF}[K] & \leftarrow \text{CDF}[K-1] + \text{cumFactor} \times \frac{N_{i,K}^{-w_j} + \gamma_i}{\sum_{i=1}^{V} (N_{i,K}^{-w_j} + \gamma_i)}
\end{align*}

we need to estimate its normalization constant. We compute the cumulative product for each $k > 1$ iterations and pass it to the next iteration of the evaluation. By doing this, we obtain a computational cost comparable to collapsed Gibbs sampling in LDA [35]. This task is illustrated in Algorithm 2. From this pseudo code, $O((2K + 1)W)$ operations are required for each Gibbs sampling draw for the whole corpus, where $W$ is the total number of words. Thus, the time complexity for each iteration is $O(KW)$ as in the case of LDA [74]. This complexity is possible due to the cascade structure of the GD distribution, which facilitates the calculation of the cumulative factor in Algorithm 2. In contrast, the time complexity of a Gibbs draw for PAM depends on the number of supertopics, $S$, and subtopics $K$ as $O(SKW)$ [56]. The general recommendation is to set $S = K/2^1$ leading to $O(K^2W)$.

\footnote{$K/2$ super-topics is the recommendation by Mallet toolbox, \url{http://mallet.cs.umass.edu/}}
2.3.3 Parameter Estimation

Previous research assumed uniform priors for the topic mixtures $\theta$ and the vocabulary distribution for topics $\phi$ without estimating them (constant values for $\alpha$ and $\gamma$) [35]. Wallach et al. in [84] concludes that parameter estimation with an asymmetric Dirichlet prior probability of topics provides an improvement in the fitting. In GD-LDA, we estimate the parameters $\alpha$, $\beta$ of the GD distribution to discover topic correlations. We estimate the parameters of the prior distribution of words given topics $\gamma$. Ideally, we should maximize the likelihood $p(w|\alpha, \beta, \gamma)$ for observations $w$ and hyper-parameters $\alpha, \beta, \gamma$ directly. Unfortunately, this distribution is intractable for this model. To solve this issue, we augment the likelihood to $p(w, z|\alpha, \beta, \gamma)$ and use Monte Carlo Expectation Maximization (MCEM) [89]. Conditional on hyper-parameters $\alpha, \beta, \gamma$, we use Gibbs sampling to estimate the posterior topic assignment distribution for each word (E-step). Then, given the expected topic assignments and words, we optimize $p(w, z|\alpha, \beta, \gamma)$ (M-step). Algorithm 3 describes these iterations.

To fit $\gamma$, we maximize the joint distribution described in Eq 2.5 conditional on the expected topic assignments, $\bar{z} = E(z|\alpha, \beta, \gamma)$, estimated from Gibbs sampling. Then we have the following optimal function:

$$
\gamma^{new} = \arg \max_{\gamma} \prod_{k=1}^{K} \frac{\Gamma(\sum_{i=1}^{V} \gamma_i) \prod_{i=1}^{V} \Gamma(\gamma_i + N_{k,i})}{\prod_{i=1}^{V} \Gamma(\gamma_i + \sum_{i=1}^{V} (\gamma_i + N_{k,i}))}
$$

(2.10)

where: $N_{k,i} = f(\bar{z})$

We follow the Newton-based approach proposed by Minka in Eqs 56-60 of [67]. Here, we have a DCM distribution to fit from $K$ observed vectors of dimension $V$. To
initialize the search, we use the method of moments based on the observed proportions

\[ p_{k,i} = \overline{N}_{k,i} / \sum_{i=1}^{V} \overline{N}_{k,i}. \]

Similarly, we estimate the parameters of the GD distribution by maximizing the joint distribution:

\[
\alpha^{\text{new}}, \beta^{\text{new}} = \arg \max_{\alpha, \beta} \prod_{j=1}^{D} \prod_{k=1}^{K-1} \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k) \Gamma(\beta_k)} \prod_{k=1}^{K-1} \frac{\Gamma(\alpha^j_k + \beta^j_k)}{\Gamma(\alpha^j_k + \beta^j_k)} \tag{2.11}
\]

where \( \alpha^j_k = \alpha_k + \overline{N}_{j,k}, \beta^j_k = \beta_k + \overline{N}_{j,k+1} + \ldots + \overline{N}_{j,K} \).

We develop a Newton-based method in Appendix A.2. To initialize the search, we use the method of moments based on the conditional beta distributions from the tree representation of the GD distribution [94]. The proportions \( p_{k,i} = \overline{N}_{k,i} / \sum_{i=1}^{V} \overline{N}_{k,i} \) are employed for this initialization.

One key component of this optimization is that each pair \( \alpha_k, \beta_k \) is optimized separately. Thus, the time complexity of this optimization is linear in time with \( K \). Parameter fitting in PAM is performed using the method of moments due the high model complexity and computational cost of the optimization [55, 56]. In addition, the number of parameters to fit for CTM and PAM is \( K^2 \) [12] [90], assuming \( K/2 \) super-topics for PAM. As a result, these methods are highly prone to over-fitting, as many parameters are being fit.

### 2.3.4 Predictive Distributions

Given the optimal parameters \((\alpha^*, \beta^*)\) obtained from Algorithm 3, and the word-topic observations \((w, z)\) the predictive distribution for document \( j \), \( \hat{\theta}_j \), is esti-
Algorithm 3 Monte Carlo EM

Start with an initial guess for $\alpha, \beta, \gamma$ and $z$

repeat

Run Gibbs Sampling using Eqs. 2.8 and 2.9

Find Expected value for topic assignments $E(z|\alpha, \beta, \gamma) = \bar{z}$

Choose $\gamma$ to maximize complete Likelihood Eqs 56-60 in [67]

Choose $\alpha, \beta$ that maximize complete Likelihood using Eq A.11

until convergence of $\alpha, \beta, \gamma$

Choose topic assignments $z^*$ with highest probability

Set $\alpha^* = \alpha, \beta^* = \beta, \gamma^* = \gamma$

\[ \text{return } \alpha^*, \beta^*, \gamma^*, z^* \]

mated as:

\[ \hat{\theta}_{jk} = \begin{cases} 
\frac{\alpha_k^* + N_{j,k}}{\alpha_k^* + \beta_k^* + \sum_{l=1}^{K} N_{j,l}}, & \text{if } k = 1 \\
\frac{\alpha_k^* + N_{j,k}}{\alpha_k^* + \beta_k^* + \sum_{l=k}^{K} N_{j,l}} \times \prod_{m=1}^{k-1} \frac{\beta_m^* + \sum_{l=m+1}^{K} N_{j,l}}{\alpha_m^* + \beta_m^* + \sum_{l=m}^{K} N_{j,l}}, & \text{if } 1 < k < K \\
\prod_{m=1}^{K-1} \frac{\beta_m^* + \sum_{l=m+1}^{K} N_{j,l}}{\alpha_m^* + \beta_m^* + \sum_{l=m}^{K} N_{j,l}}, & \text{if } k = K 
\end{cases} \] (2.12)

for the topics $k = 1 \ldots K$ and the documents $j = 1 \ldots D$.

The predictive distribution for the probability of words given topics, $\hat{\phi}_k$ is
estimated as follows [35]:

$$\hat{\phi}_{ki} = \frac{N_{i,k} + \gamma_i^k}{\sum_{t=1}^{T} (N_{i,k} + \gamma_i^t)}$$  \hspace{1cm} (2.13)

Notice that the probability of topics $\hat{\theta}_j$ is document dependent. On the other hand, the probability of words given their topic $\hat{\phi}_k$ is topic dependent. This dependency implies that for an unseen document, we need to estimate its predictive distribution of topics while the probability of words given their topics remains the same.

### 2.4 Experimental results

#### 2.4.1 Validation

The primary challenge to validate statistical topic models is the lack of reliably observed topic labels for each word in the corpus. The standard approach is to estimate the likelihood of held-out data [85]. In addition, model performance in a supervised task has also been used [98]. Here, we validate the GD-LDA performance addressing these two forms. We estimate the likelihood of completely unseen documents using Empirical Likelihood, and we compare the performance of topic models in the ad hoc Information Retrieval as developed in [90].

#### 2.4.1.1 Empirical Likelihood (EL)

There has been a debate about the evaluation of topic models with methods ranging from the harmonic mean of complete a likelihood for topic assignments to perplexity and empirical likelihood.
As discussed in [85], perplexity falls into the category of document completion where a portion each document must be observed to estimate the likelihood of the remaining content. Similarly, a "left to right" evaluation has been proposed to determine the probability of words in a test document incrementally [85]. For validation, we use Empirical Likelihood (EL) criterion. Our intent is to predict the likelihood of fully unseen documents.

We do not use perplexity or "left to right" evaluation because they are based on the word order inside the document, in contrast to "bag of words". In addition, EL has been shown to be a more pessimistic approximation for the probability of held-out documents than "left to right" method [85].

We estimate EL as described in [56]. Here, we generate a set of pseudo-documents, $\theta_s$ using the estimated prior topic distribution of the model being tested using training data. Then, a word distribution is determined for each $\theta_s$ based on $\hat{\phi}$. This value is used to estimate the probability of seeing the test set. We define this process using Algorithm 4. Here, $D_t$ represents the number of documents for a test set, $d_t$ is the document $t = 1, \ldots, D_t$, $w_t$ are the $N_t$ words of $d_t$, and $N_s$ is the number of pseudo-documents given model $M$.

We use the generative approach based on the conditional Beta distributions of the tree representation for GD as in Fig 2.1.

The main limitation of EL estimation is that the number of pseudo-documents should be sufficiently large to cover the parameter space of $\theta$ given the trained model. Then, $N_s$ determines the accuracy of the approximation.
Algorithm 4 Empirical Likelihood Estimation for model $M$

for $s \leftarrow 1$ to $N_s$ do

Sample document $d_s$ with mixture $\theta_s$ given model $M$

Find $p(w = i|\theta_s, \hat{\phi}, M) = \sum_{k=1}^{K} \theta_{sk} \hat{\phi}_{ki}$ for $i = 1, \ldots, V$

end for

Set $\log EL = 0$

for $t \leftarrow 1$ to $D_t$ do

Find $p(d_t|d_s, M) = \prod_{w_t=1}^{N_t} p(w_t|\theta_s, \hat{\phi}, M)$

Find $p(d_t|M) = \frac{1}{N_s} \sum_{s=1}^{N_s} p(d_t|d_s, M)$

Update $\log EL = \log EL + \log(p(d_t|M))$

end for

return $\log EL$

2.4.1.2 Ad hoc Retrieval

Information Retrieval represents a hard problem where the gains of new models are often small or nil. This application represents a real test of the power of our approach. We compare the performance of different topic models in ad hoc Information Retrieval (IR). We use the approach proposed in [90] and incorporate our model by replacing the LDA model described in the method. Based on the predictive distribution estimators for topics $\hat{\phi}$, and for each document $\hat{\theta}_j$, we provide a topic-based language model for each document, $P_{TM}(w|\theta_j)$, as follows:

$$P_{TM}(w|\theta_j, \hat{\phi}) = \sum_{k=1}^{K} P(w|z = k, \hat{\phi}_k) P(z = k|\theta_j)$$  \hspace{1cm} (2.14)
This is augmented with the maximum likelihood estimate for the language model based on document terms \(D_j\), \(P_{ML}(w|D_j)\), and for the language model based on the corpus \(C\), \(P_{ML}(w|C)\), leading to:

\[
P_{IR}(w|D_j, \hat{\theta}_j, \hat{\phi}) = \lambda \left( \frac{N_j}{N_j + \mu} P_{LM}(w|D_j) + \frac{\mu}{N_j + \mu} P_{ML}(w|C) \right) + (1 - \lambda) P_{TM}(w|\hat{\theta}_j, \hat{\phi})
\]

(2.15)

where \(\mu\) is a smoothing parameter, \(N_j\) is the number of words in document \(j\), and \(\lambda\) is a parameter for the linear combination. Therefore, the ranking function for query \(Q\) with terms \(q\) is given by:

\[
P(Q|D_j, \hat{\theta}_j, \hat{\phi}) = \prod_{q \in Q} P_{IR}(w = q|D_j, \hat{\theta}_j, \hat{\phi})
\]

(2.16)

We use the parameter values: \(\mu=1000\), \(\lambda=0.7\). These are the values recommended in [90].

### 2.4.2 Experimental Settings

We use 4 different data sets to test our model. NIPS conference papers data set\(^2\) which contains long documents (8-10 pages). We also test on two news data sets, NYT, and APW obtained from TREC-3. These collections contain shorter documents (1 page) with different vocabulary size. A fourth dataset, OHSUMED from TREC-9, consists of abstracts from medical papers. In contrast to the other data sets, the number of documents is much larger (more than ten times). Table 2.1 shows the features of these datasets. We remove standard stop words and perform stemming in all datasets.
Table 2.1: Features of the datasets analyzed. Mean document includes the 95% interval

<table>
<thead>
<tr>
<th>Data set</th>
<th>NIPS</th>
<th>NYT</th>
<th>APW</th>
<th>OHSUMED</th>
</tr>
</thead>
<tbody>
<tr>
<td># documents</td>
<td>1840</td>
<td>5553</td>
<td>14657</td>
<td>196404</td>
</tr>
<tr>
<td># unique terms</td>
<td>13649</td>
<td>11229</td>
<td>18471</td>
<td>38900</td>
</tr>
<tr>
<td>Mean doc. length</td>
<td>$1322 \pm 274$</td>
<td>$274 \pm 132$</td>
<td>$169 \pm 76$</td>
<td>$186 \pm 36$</td>
</tr>
</tbody>
</table>

Table 2.2: Number of Gibbs samples per EM iteration

<table>
<thead>
<tr>
<th>MCEM iteration</th>
<th>1-4</th>
<th>5-6</th>
<th>7-8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burn-in Samples</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>Gibbs Samples</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>500</td>
<td>700</td>
</tr>
</tbody>
</table>

We test 4 variants of the GD-LDA algorithm. In two of these cases, we fix the value of $\gamma$ and optimize the parameters $\alpha$ and $\beta$ as in Algorithm 3. Here, we use two variants: 1) using the empirical expected topic assignments $\bar{z}$ [89], *GD-LDA Fixed Gamma Mean*; and 2) using the empirical mode $z^*$ [24] in the M-step, *GD-LDA Fixed Gamma Max*. In the other two cases, we optimize all the three parameters ($\alpha, \beta$ and $\gamma$) based on the expected topic assignments, *GD-LDA Mean*, and the topic assignment mode *GD-LDA Max*.

We compare GD-LDA, LDA with parameter optimization, CTM, and PAM. For LDA, we use our implementation. We follow the collapsed Gibbs sampling approach [35], with an asymmetric Dirichlet prior distributions for both, the probability of words given a topic, and for the probability of topics. These parameters are optimized as discussed by authors in [23]. For optimization, we follow the Newton-based approach proposed by Minka in Eqs 56-60 [67]. For both GD-LDA and LDA, we set a maximum of 1,000 Newton iterations for parameter optimization. The schedule of the Gibbs sampling is detailed in Table 2.2. We initialize the values $\alpha_k = 2/K$ for $k = 1, \ldots, K-1$,

\[\text{Available at: http://cs.nyu.edu/~roweis/data.html}\]
and \( \beta_{K-1} = 2/K \) in GD-LDA to obtain the particular case of Dirichlet distribution. We start with LDA as a prior model and allow the data to adapt the model in the MCEM iterations to the more general GD. For CTM\(^3\), we use the default settings with parameter estimation: a maximum of 1,000 EM iterations with convergence of \( 10^{-5} \) and maximum of 20 variational iterations with a convergence rate of \( 10^{-6} \). For PAM\(^4\), we use 1000 Gibbs samples and \( K/2 \) super topics. This implementation supports multi-threading; to enable a fair comparison we use only one thread. We modify the application to obtain the fitted parameters since they are not provided by default.

For EL estimation, we perform 10-fold cross-validation and we sample \( N_s = 10,000 \) pseudo documents.

### 2.4.3 Results

#### 2.4.3.1 Qualitative Results

Fig 2.3 shows a qualitative example of GD-LDA in the NYT dataset. The figure represents the empirically estimated probability of topics for a subset. Each box represents a topic and the terms displayed are the ones with highest posterior predictive probability \( \hat{\phi}_k \). Given the fitted GD-LDA model, we calculate the point estimate of the total probability for each topic (in blue) and its conditional probability given a parent node (red). Here, dependent on the current node, we move deeper into the tree or observe a word from the topic at that level. As we move down into the tree, the

\(^3\)We use the CTM implementation provided by Blei at [http://www.cs.princeton.edu/~blei/ctm-c/](http://www.cs.princeton.edu/~blei/ctm-c/).

\(^4\)We use the implementation provided by Mallet [http://mallet.cs.umass.edu/](http://mallet.cs.umass.edu/).
Figure 2.3: Qualitative example of GD-LDA for the NYT data set with 20 topics. Probability of topics: marginal probability as defined by Eq A.2 (red), conditional probability given the parent node as defined by Eq A.6 (blue).
conditional probability of picking the left-hand topic increases since the probability mass of the remaining topics is less. This is a desirable property that facilitates dimensionality reduction. If this probability is large compared with the probability of exploring further the tree, we can discard the remaining tree.

Fig 2.4 shows some of the positive and negative correlations between different topics inferred by GD-LDA. We observe that *Health* is positively correlated with *Technology* and *Financial*. *Y2K* is positively correlated with *Technology* and *Arts* but negatively correlated with *Foreign Politics*. Note that by default LDA assumes neg-
Figure 2.5: Qualitative comparison of the topic distribution for two documents from the NYT data set. (a) From left to right: PAM, CTM, and GD-LDA (b) From left to right: GD-LDA and CTM

ative correlations. Thus, the primary value of correlated topic models is to discover positive correlations. Fig 2.5(a) shows the decomposition of a sports document from the NYT dataset into topics using PAM with 10 super-topics and 20 sub-topics, CTM and GD-LDA with 20 topics. This figure also shows that CTM and GD-LDA assign a high probability to a single topic (sports). On the other hand, PAM provides three
sub-topics and most of the super-topics with a significant probability mass. This mass probability distribution is not desirable since an important objective with topic modeling is to cluster documents based on the topic mixture. Fig 2.5(b) shows a comparison between CTM and GD-LDA for a document about the Y2K problem and airport functionality. We observe that GD-LDA provides better segmentation of the document based on the topics proportions and content.

In addition, we calculate the distribution of the number of significant topics in a document. We rank the topics based on their probability mass in each document. Then, we estimate the number of topics which accounts for 95% of this probability mass. Fig 2.6 shows the distribution of this number of topics inferred by LDA, CTM, PAM,
and GD-LDA. For PAM, we consider the number of sub-topics. Here, we observe that CTM favors a high number of topics for each document. This fact introduces noise when the goal is to characterize documents as in Information Retrieval. In PAM, the number of topics per document is highly dependent on the number of supertopics used. In order to handle topic correlation sparsity, PAM prunes the relationships between a super topic and subtopics [56]. This reduces the amount of correlations that can be modeled. In contrast, GD-LDA favors a smaller number of topics per documents and without any constraint in the inference. The fundamental property of this behavior is the cascade structure of the distribution. As discussed in section 2.2.2, this validates the intuition of few significant topics for natural language documents and sparse correlations empirically.

2.4.3.2 Empirical Likelihood Results

Fig 2.7 shows EL estimations for the four variants of GD-LDA, CTM, PAM and LDA with asymmetric prior and parameter optimization in the four datasets. For the NIPS data set, LDA is not shown since its predictive likelihood is extremely small. Similarly, PAM performance for the OHSUMED data set is not indicated for the same reason. The CTM model would not run for the OHSUMED data set (mid-size data set) based on the number of documents and vocabulary size.

We observe that optimizing $\gamma$ has a small impact as more data becomes available. Conceptually, a prior distribution represents the prior knowledge with a given sample size. Consequently adding more data decreases the effects of the prior informa-
Figure 2.7: Mean per-document log-Likelihood for (a) NIPS, (b) NYT, (c) APW and (d) OHSUMED datasets as a function of the number of topics.

We then compare the use of the topic assignments, $\bar{z}$ (posterior mean), and the topic assignment mode, $z^*$ (maximum a posteriori MAP). We find that *GD-LDA mean* performs better when the number of documents and the vocabulary size are relatively small (NIPS data set). In contrast, *GD-LDA Max* shows the highest performance for the other datasets, Fig 2.7(b)-(d). In particular, this method performance is clearly superior for all the topics when tested for the largest data set (OHSUMED). *GD-LDA Mean* requires fewer topics to achieve superior performance compared to LDA, CTM, and PAM. Moreover, EL decreases more smoothly than these methods. It also remains relatively constant when the number of topics attains a high value.

A significant difference between the datasets analyzed is the average document
Table 2.3: Results of ad hoc IR using GDLDA, LDA and PAM models using \( K \) topics for the OHSUMED data set

<table>
<thead>
<tr>
<th>Model</th>
<th>( K )</th>
<th>P@10 ( \pm )</th>
<th>MAP ( \pm )</th>
<th>DG ( \pm )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD-LDA</td>
<td>50</td>
<td>0.502( \pm )0.06</td>
<td>0.496( \pm )0.06</td>
<td>0.790( \pm )0.07</td>
</tr>
<tr>
<td>LDA</td>
<td>50</td>
<td>0.470( \pm )0.06</td>
<td>0.470( \pm )0.06</td>
<td>0.780( \pm )0.08</td>
</tr>
<tr>
<td>PAM</td>
<td>50</td>
<td>0.434( \pm )0.05</td>
<td>0.439( \pm )0.05</td>
<td>0.671( \pm )0.04</td>
</tr>
<tr>
<td>GD-LDA</td>
<td>75</td>
<td>0.470( \pm )0.06</td>
<td>0.491( \pm )0.06</td>
<td>0.742( \pm )0.04</td>
</tr>
<tr>
<td>LDA</td>
<td>75</td>
<td>0.470( \pm )0.06</td>
<td>0.461( \pm )0.06</td>
<td>0.741( \pm )0.04</td>
</tr>
<tr>
<td>PAM</td>
<td>75</td>
<td>0.431( \pm )0.06</td>
<td>0.438( \pm )0.05</td>
<td>0.651( \pm )0.03</td>
</tr>
</tbody>
</table>

Length and number of unique terms. The worst performance of CTM, as well as PAM, is found in the case of APW data set. This data set has a larger vocabulary size, and significantly shorter documents than NIPS data set, where both CTM and PAM show a similar performance to that of GD-LDA Mean. This suggests over-fitting by CTM where a full topic covariance matrix is estimated, and by PAM where a matrix of \( K \) subtopics by \( K/2 \) super-topics is estimated. In addition, PAM uses the method of moments, instead of optimization, for parameter estimation. This is a limitation of PAM when EL based evaluation is performed since the fitted parameters do not optimize the likelihood. As discussed in section 2.4.3.1, CTM favors a larger number of topics for each document than the other methods. This behavior introduces noise in EL because correlations are sparse, and only a few topics are present in natural language documents.

2.4.4 Adhoc Document Retrieval

We show the application of GD-LDA in ad hoc IR using the OHSUMED data set. As discussed above, we use the approach from [90] as a benchmark for comparison.
We train the *GD-LDA Max* using $K = 50$ and $K = 75$ topics, and compare its performance with LDA and PAM. We use standard IR measures: Precision at 10 (P@10), Mean Average Precision (MAP) and Discounted Gain (DG), to compare these methods. The OHSUMED dataset is considered to be a medium size one in the IR literature. This dataset contains 63 topical queries with 35,000 relevant labels.

We have not modified the retrieval model of [90] to exploit the power of the new GD-LDA model. Despite this, the performance improvement for all the measures is significant as we observe in Table 2.3. Here, the best performance is for $K = 50$ topics, where EL estimation peaks in Fig 2.7(d). For this case, there is an agreement between the ad-hoc IR and EL performance. Respect to LDA, GD-LDA shows improvement of: 6.3\% for P@10, 5.5\% for MAP, and 1.1\% for DG. Note that PAM shows lower performance than LDA in IR for all the measures. This result is consistent with the IR performance of PAM reported in [98].

### 2.4.4.1 Computational Cost Comparison

One advantage of GD-LDA over CTM and PAM is that its computational complexity is linear in the number of topics. This is not the case for CTM and PAM, which scale quadratically with the number of topics. As discussed in section 2.3, GD-LDA should add minimal computational cost to LDA.

Fig 2.8 shows the computational time in minutes to fit the model for LDA, CTM, PAM and GD-LDA in the datasets we consider. A non-linear increase in time is observed after 50 topics for CTM in NYT and APW datasets, and after 80 topics
in NIPS data set. We find that the computational cost of PAM grows quadratically after 20 topics. In general, the computational cost of GD-LDA is comparable to that of LDA and is less than CTM and PAM. This is a significant advantage since GD-LDA provides a more flexible model structure when compared with LDA. Moreover, variances and some covariances are modeled more efficiently, and with minimal increase in computational cost.

This property makes GD-LDA more suitable than CTM or PAM for larger-scale applications, such as IR.
2.4.4.2 Choosing the Number of Topics

An open problem is how to select the optimal number of topics to train a model. In general, an exhaustive experimentation needs to be performed to choose the optimal number of topics. Due to the tree structure of the GD distribution, GD-LDA tends to accommodate the most relevant topics, based on probability mass, at the upper levels of the tree.

Fig 2.9 shows the cumulative probability vs. the number of topics of the expected topic mixture of documents given the fitted parameters for GD-LDA and CTM. We observe that GD-LDA favors a smaller number of topics.

Notice that, after a certain number of topics, the expected contribution of the remaining ones is not significant. This prevents GD-LDA from over-fitting. When comparing GD-LDA with CTM, we observe that CTM favors uniform topic mixtures.
in each document. Thus, if we fit CTM for a larger number of topics, we would find
the same linear behavior as seen in graphs from Fig 2.9. This prevents CTM from
discarding any topic easily or suggesting an optimal topic range as opposed to GD-
LDA. In addition, this behavior makes CTM less efficient in handling over-fitting.

2.5 Discussion

We have introduced the use of the GD distribution in probabilistic topic mod-
eling. The advantages of the GD over the Dirichlet distribution and the benefits, when
compared with the estimation of the full covariance matrix in CTM, have been described.

The apparent constraints on covariances in GD result in modeling better sparse
topic correlations in natural language documents, as our empirical validation indicates.
This results in better performance in empirical likelihood and IR measures. We have
developed an efficient Gibbs sampling model that uses the conjugacy property of GD
with the Multinomial distribution. We have demonstrated that the running time of
GD-LDA is comparable to LDA and less than CTM and PAM. This provides a model
computationally competitive and with better performance than these methods.

We have shown that the impact of optimizing the vocabulary parameter $\gamma$
decreases when the vocabulary size and the number of documents in the corpus are
large.

Due to the tree structure of the GD distribution, GD-LDA proves to be pow-
erful in handling over-fitting with a large number of topics as its performance remains
fairly high even when we increase the number of topics. This is not the case for CTM, PAM, and LDA. As a consequence, we can reduce the number of topics, by using the conditional probability of the remaining topics when we are moving down into the tree. A natural extension of the model is to allow the tree structure to be fitted. A direction of improvement is to modify the Dirichlet tree to have a comparable notion of subtopics and super topics as in PAM with

We have shown that the use of GD-LDA in ad-hoc IR increases the performance significantly, in contrast to earlier incorporations of topic models.

Future research would involve making GD-LDA scalable at the website level. To perform this task, we would need to estimate the topic mixture and update the model parameters in a distributed form. However, the primary constraint of fitting GD-LDA in this scenario relies on the sequential nature of Gibbs Sampling. The authors of [69] suggest that relaxing the subsequent assumption in the Gibbs Sampling does not affect the final results of fitting other topic models significantly.

Therefore, we can fit the parameters of the model in a distributed form by dividing the corpus into different data sets that would be sent to cluster nodes. In each node, we would estimate the topic label of each word found in each document in the data set assigned to each cluster node using an initial parameter set. Then, we would obtain sufficient statistics from all the documents attached to each node (Topic Labels Counts) that later would be used to update the parameters of the model globally.
Chapter 3

Application of Topic Models in

Relevance Feedback

3.1 Introduction

Relevance feedback has been studied extensively in Information Retrieval as a form of incorporating feedback from the user to refine the results retrieved. The authors in [97] concluded that negative feedback is also valuable to improve the ranking. However, the need to capture the broader context in difficult queries is still a challenge. The authors in [51] have showed that including global features and using clusters can improve the retrieval performance significantly. Thus, statistical topic modeling provides a robust and an automatic method to incorporate context to the user feedback.

Previous approaches have used statistical topic models to represent documents according to their latent topic content and use this representation in information re-
Authors in [90], use topics as a form of smoothing the Language Model used in retrieval. However, this approach does not address the incorporation of relevance feedback.

Recent work from [2] explores the use of topics as a form to perform the query expansion for relevance feedback. However, this action might make the query noisier because the top topic terms might not contribute to a better discrimination of the relevant documents. In addition, these terms might not be distinctive across different topics.

We propose to include the topic information as feedback using the document topic mixture instead of the document word mixture. We first estimate the topic mixture for each document in the corpus using LDA and save it as metadata. Given an initial query, we use a standard retrieval engine, Language Models for this case, to show the first set of documents to the user and obtain relevance judgments. Then, we assume that topic mixtures for feedback documents are observed. We then define two latent Dirichlet distributions: one for relevant documents and another for non-relevant documents. We fit these distributions iteratively, by finding a sufficient statistic and maximizing the likelihood of observing this statistic. To score the documents, we use Bayesian Logistic Regression. This function results in a very efficient scoring function and incorporates the benefits of active learning. Under this model, we incorporate positive and negative feedback, and context based on topics in the interaction without changing the query. We also provide efficient updates of the latent distributions based on topics.

This chapter is organized as follows: in section 3.2, we present the methodology...
to use the topic mixture in relevance feedback. In section 3.3, we present results of the
method. Finally, a discussion and conclusion is presented in 3.4.

3.2 Topic Models in Relevance Feedback

In this section, we describe how we incorporate the topic mixture of feedback
documents as a global measure in contrast to query expansion. To achieve this, we
estimate the topic mixtures of the documents, $\theta_i$, for $K$ topics in the corpus using LDA
off-line. Given an initial ranking, the user provides relevance feedback which is used to
fit the latent relevant/non-relevant distributions of topic mixtures. Thus, we assume
two Dirichlet distributions: one for the relevant set of documents $\alpha_R$, and one for the
non-relevant documents $\alpha_{\bar{R}}$. Therefore, we have:

$P(\theta_i | \alpha_R) \sim \text{Dirichlet}(\alpha_R) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_{k,R})}{\prod_{k=1}^{K} \Gamma(\alpha_{k,R})} \prod_{k=1}^{K} \theta_{i,k}^{\alpha_{k,R} - 1}$

for $\alpha_R$ and $\alpha_{\bar{R}}$. Then, we calculate the log-probability of the document being generated
by those distributions:

$\log P(\theta_i | \alpha) = \log \Gamma\left(\sum_{k=1}^{K} \alpha_k\right) - \sum_{k=1}^{K} \log \Gamma(\alpha_k) + \sum_{k=1}^{K} (\alpha_k - 1) \log(\theta_{i,k})$

We denote these scores as $PR_i$ and $P\bar{R}_i$ respectively. To update the latent
distributions of the relevant $\alpha_R$ and non-relevant topics $\alpha_{\bar{R}}$, we use the topic content
from the documents. Here we used the documents labeled by the user as relevant, $\theta_R$, and
non-relevant, $\theta_{\bar{R}}$, after each interaction.

The Dirichlet distribution guarantees a unique maximum when the Maximum
Likelihood (ML) is estimated for $\alpha$. Moreover, a sufficient statistics, $SS$, can be determined to update this distribution as more observations are available. We can update $SS$ efficiently without keeping previous document feedback. The initial value of the sufficient statistic $SS^{(0)}_{k,R}$ for the relevant topic $k$ and its update from the interaction $j$ is described by:

$$SS^{(0)}_{k,R} = \frac{1}{N^{(0)}_R} \sum_{i \in R_0} \log \theta_{i,k}$$

$$SS^{(j)}_{k,R} = \frac{N^{(j-1)}_R}{N^{(j)}_R} SS^{(j-1)}_{k,R} + \frac{1}{N^{(j)}_R} \sum_{i \in R_j} \log \theta_{i,k}$$

where $N^{(j)}_R = N^{(j-1)}_R + |R_j|$, and $|R_j|$ is the total number of relevant documents at the $j$-th interaction. Given $SS^{(j)}_{R}$ and $SS^{(j)}_{\bar{R}}$, we use the method proposed in [67] to calculate the ML estimator for $\alpha^{(j)}_R$ and $\alpha^{(j)}_{\bar{R}}$.

In addition to these distributions, we use the topic-based Language Model $P_{TW}$ for document $i$ as follows:

$$P_{TW,i}(w|\theta_i, \hat{\phi}) = \sum_{k=1}^{K} P(w|z = k, \hat{\phi}_k) P(z = k|\theta_i)$$

where $\hat{\phi}$ are the word mixture for the topics obtained from LDA. Thus, the score $S_{TW,i}$ for query $Q$ with terms $q$ is defined as:

$$S_{TW,i} = \prod_{q \in Q} P_{TW,i}(w = q|\theta_i, \hat{\phi})$$

To combine the scores from the latent relevant/non-relevant topic mixtures and the topic-based Language Model, we use the Bayesian Logistic Regression approach [97].
Let $y_i = \{+1, -1\}$ be the relevant/non-relevant label for document, we have the score function:

$$P(y_i | \beta, \mathbf{d}_i) = \frac{1}{1 + \exp(-\beta^T \mathbf{d}_i y_i)}$$

where $\mathbf{d}_i$ is the feature vector scores: $PR_i, P\bar{R}_i, S_{TW,i}$. $\beta$ is a parameter vector assumed to be normally distributed and updated in a Bayesian form. Here, the distribution of $\beta$ from the $j$-th iteration is taken as a prior distribution for the next iteration. To approximate the posterior distribution we use the Laplace approximation as discussed in [97].

3.3 Results

We test our method using the OHSUMED dataset that consists of 196,000 medical abstracts and 3,506 relevance labels for 63 queries from the Document Filtering Track from TREC 4. As suggested in the track, we assume unobserved labels as non-relevant. We fit the LDA model using $K = 50$ topics, which is the number of topics with the highest performance based on Empirical Likelihood. To test the impact of topic information, we use standard Language Model (LM) with Dirichlet smoothing described in [90] as a baseline. This score is used with Bayesian Logistic Regression. We test 3 variants of the model and the baseline: LM as a baseline; LM+$S_{TW}$; LM+$S_{TW}$+$PR$; LM+$S_{TW}$+$PR$+$P\bar{R}$. We calculated the initial ranking using LM and asked for feedback until we have at least one relevant and one non-relevant document. We use ten feedback documents and estimate precision at 10 (P@10), Mean Average Precision (MAP),
Table 3.1: Results of Topic feedback using 50 topics in the OHSUMED dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>P@10</th>
<th>MAP</th>
<th>DiscGain</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>0.3968</td>
<td>0.4286</td>
<td>0.5660</td>
</tr>
<tr>
<td>LM+STW</td>
<td>0.4206</td>
<td>0.4557</td>
<td>0.6315</td>
</tr>
<tr>
<td>LM+STW+PR</td>
<td>0.2968</td>
<td>0.3307</td>
<td>0.5590</td>
</tr>
<tr>
<td>LM+STW+PR+P̄R</td>
<td>0.4698</td>
<td>0.5141</td>
<td>0.6580</td>
</tr>
</tbody>
</table>

and Discounted Gain (DiscGain). There are two relevance level labels available in the dataset, \{1, 2\}, that are assumed equally, \{+1\} for P@10 and MAP. However for DiscGain, we use both labels in the evaluation.

Table 3.3 shows the results for the variants tested. We observe that the LM+STW performs better than the baseline. This score is similar to the LDA-based retrieval proposed in [90] but the value of the linear combination parameters \(\beta\) is fitted based on the feedback as opposed to a corpus-wide parameter. When we incorporate only the score from the relevant distribution of topics PR, the performance decreases. However, when we include the score for the non-relevant distribution P̄R, the performance is the highest. This fact shows the value of negative feedback reported previously in [97]. We notice that the combination of PR and P̄R is equivalent to the log of the likelihood ratio test (probabilistic ranking principle) weighted by \(\beta\). This also explains why both scores should be included in the model.

We observe that the combination of the four scores improves the general performance by: **11.6%** P@10, **12.8%** MAP, and **4.6%** in the Cumulative Discounted Gain with respect to topic-based language model (LM+STW). This demonstrates the power of statistical topic modeling in relevance feedback.
3.4 Discussion

In this chapter, we have presented a method to incorporate the statistical topic information in relevance feedback without changing the query. Results show that including the mixture of topics in relevance feedback improves the performance by pruning the search space while adding context to the query.

As future work, we plan to incorporate a policy to decide when to update the parameters of the relevant and non-relevant topic distribution optimally.

The use of the non-relevant document distribution allows us to reduce the document search space by reducing the effect of nonrelevant content within the relevant labeled documents.

Although the documents in the corpus are exchangeable, the performance of the model is highly dependent on the order that the documents are retrieved and how discriminative each document is. The method presented in this chapter is based on reducing the variance in the parameters (exploration). Including a maximization of relevance in the least number of iterations (exploitation) in the ranking score would decrease the dependence on the order of retrieved documents. The correct combination of both measures is still an open research path.
Chapter 4

Application of Topic Models in
Healthcare related Document Retrieval

4.1 Introduction

Today an increasing number of healthcare related documents are available on
the web. Widely known pages such as WebMd and Mayo Clinic describe detailed
symptoms and treatments for different diseases. Patients often search for treatments
and disease care options that are available after they are discharged from the hospital.
However, the search in this domain poses challenges such as synonyms and related
terms for the same disease and treatment that prevent traditional keyword search from
being effective. An ontology tree is often used to address these challenges. Here, the
query is expanded to include related terms in the search [17]. In this framework, the
documents are decomposed into concepts [65]. However, performing this task with web
data is unfeasible due to the high computational cost of the annotation process and the heterogeneity of the corpus when a fixed ontology is used.

We propose a method based on Statistical Topic Models to provide global information about documents. These methods provide an unsupervised clustering framework that enables us to increase the weight of related words without including them explicitly in the document or the query. In addition, we describe a method to incorporate noun phrases in the vocabulary without annotating the entire corpus using Natural Language Processing tools. In this paper, we explore the effectiveness of using patient discharge summaries to provide relevant context to the query and to improve the retrieval performance. Here, we exploit the fact that patients read their discharge summaries and pose related queries afterward.

 Traditionally, healthcare document retrieval relies on concepts such as symptoms, medications, and diseases. Current approaches are based on a concept and ontology-based information retrieval [65]. These methods often require the documents to be annotated using a list of conceptual terms from ontologies trees or taxonomies [65, 17, 18]. Then string matching and rule based retrieval techniques are employed to rank documents. To improve the document score given a query, bag-of-words retrieval methods in the concept domain have been proposed. Castells et. al. [17] introduced a vector space model in ontology-based retrieval as a form to improve the precision. Other approaches combine the ontology and the bag-of-words based scoring function to discriminate between concepts [65]. The main drawback of these methods is the annotation task. This process requires documents to be annotated using a fixed taxonomy.
This task is not feasible when using multi-source web documents due to the discrepancy between the scientific nature of the ontologies and the common language of the web documents, and the size of the corpus. In addition, the annotated text does not reflect the intensity of a concept inside the document and often requires disambiguation [45]. Annotators usually write the different concepts inside the document without indicating which are the most dominant ones. To alleviate all these issues, the query is often expanded using related concepts obtained from the ontology tree instead of annotating the document[79, 102]. Other approaches use Relevance Feedback based methods to disambiguate different concepts inside the document[65].

In contrast, we propose an unsupervised method that exploits the clustering power of Statistical Topic Modeling methods based on bag-of-words. This technique uses the term co-occurrence to establish semantic relations to add context to the query. We also explore if the use of contextual text information extracted from patient’s discharge summaries improves the retrieval performance.

This chapter is organized as follows: in section 4.2 we present a review of topic Models. In section 4.3, we introduce the method we used to extract Noun Phrases. We depict the feature extraction process in section 4.4. In section 4.5, we present the highlights of the retrieval method we use to rank the documents. Results are shown in section 4.6. Finally a discussion and conclusion is provided in section 4.7.
4.2 Statistical Topic Modeling

Statistical Topic Modeling is an unsupervised learning technique that allows us to extract latent topics from a document corpus. The central idea is that the extracted topics provide a global context, which cannot be obtained by using independent word counts as in the standard bag-of-words models. For the current problem, we use the Generalized Latent Dirichlet Allocation (GD-LDA) topic model [16]. This technique has been shown to improve the document retrieval by constraining the number of covariances among the topics. In this model, each document is defined as a statistical mixture of topics, and each topic is described as a mixture of words. The most dominant topics in the document are those with the highest probability. Similarly, the higher the probability of a given term the more relevant the term is to the latent topic. For model fitting, we follow a Gibbs sampling based approach as detailed in [16].

We estimate the prior distribution parameters that optimize the augmented likelihood which is defined as follows:

\[
p(w, z|\alpha, \beta, \gamma) = p(w|z, \beta)p(z|\alpha, \gamma) = \prod_{i=1}^{N_j} \prod_{j=1}^{J} \prod_{k=1}^{K} \phi_{k,i} \times \theta_{j,k} \tag{4.1}
\]

where \( w \) is the set of document words in the corpus, \( z \) are the topic labels for each word, \( \alpha \) and \( \gamma \) are the prior distribution parameters of the topic mixture. Similarly, \( \beta \) is the parameter vector of the prior distribution of word terms in the vocabulary given a topic. \( \theta_{j} \) is the topic mixture of the document \( j \) and \( \phi \) is the word mixture for each latent topic \( k \). In this framework, we need to fix the number of topics \( K \). We fit the model using
10% of the documents in the corpus selected randomly. Once we estimate the prior distribution parameters, we extract the topic content for the remaining documents by sampling the labels for each word in the document given predicted the prior distribution of the word mixture \( \hat{\phi} \). Our goal is to depict a real scenario where only a subset of the corpus is used to train the model. We estimate the topic content \( \hat{\theta}_j \) for each document using the following equation:

\[
\hat{\theta}_{jk} = \begin{cases} 
\frac{\hat{\alpha}_k + \sum_{l=1}^{K} \hat{N}_{j,l}}{\hat{\alpha}_k + \hat{\beta}_k + \sum_{l=1}^{K} \hat{N}_{j,l}}, & \text{if } k = 1 \\
\frac{\hat{\alpha}_k + \sum_{l=1}^{k} \hat{N}_{j,l}}{\hat{\alpha}_k + \hat{\beta}_k + \sum_{l=k}^{K} \hat{N}_{j,l}} \times \prod_{m=1}^{k-1} \frac{\hat{\beta}_m + \sum_{l=m+1}^{K} \hat{N}_{j,l}}{\hat{\beta}_m + \hat{\beta}_m + \sum_{l=m}^{K} \hat{N}_{j,l}}, & \text{if } 1 < k < K \\
\prod_{m=1}^{k-1} \frac{\hat{\beta}_m + \sum_{l=m+1}^{K} \hat{N}_{j,l}}{\hat{\alpha}_m + \hat{\beta}_m + \sum_{l=m}^{K} \hat{N}_{j,l}} & \text{if } k = K
\end{cases}
\]

for the topics \( k = 1 \ldots K \) and the documents \( j = 1 \ldots J \). The value of \( \hat{N}_{j,k} \) is the topic label counts obtained from Gibbs sampling. We assign a topic label \( z \) to each observed word inside the document using the following expression:

\[
p(z_{wj} = k|z_{-wj}, \alpha, \beta, \gamma) = \hat{\phi}_{w,j} \times \hat{\theta}_{j,k} \]

where \( -wj \) represent the estimation without the current word \( w \) in document \( j \). We calculate the topic mixture for each document and save it as metadata to be used later in the retrieval task.
4.3 Noun Phrases Extraction

Noun phrases provide relevant information about diseases and treatments that are not accounted for when we use the bag-of-words scheme. Several conditions and medications are often identified by two or more terms. There are also several word combinations that can be considered as stop words (i.e., patient name, physician name) that are removed when we index the corpus. However, annotating the entire corpus is not a feasible task due to the heterogeneity and size of the corpus. In addition, when we annotate a document, several concept candidates can match the selected text. This fact challenges the performance of the entity and concept-based retrieval due to the ambiguity of the resulting annotations. To alleviate this problem, we extract the most common noun phrases that depict healthcare related content by annotating a small part of the corpus using MetaMap [5] extraction tool. This tool uses Natural Language Processing techniques and the UMLS ontology tree [13] to annotate documents. We set up Metamap to return candidates with drug, disease and procedure information by observing their semantic type with the acronym disambiguation option.

After completing the annotation process, we select the noun phrases formed by 2 or more terms that have healthcare related content. Then, we construct a unique identifier that replaces each of the extracted noun phrases inside the document. With this process, we include a typical set of nouns with clinical content used in the web pages without annotating the entire corpus.
4.4 Feature Extraction from Discharge Summaries

Discharge summaries provide additional information to discover the context of the user who poses the query. However, only a fraction of the discharge summary is related to the query. These documents are often formed by a collection of unrelated events. To effectively add context to the query, we only take into account those paragraphs with significant relation to the query. We determine this relationship in an unsupervised manner by comparing the query terms and paragraph term distributions using KL divergence and selecting the ones with the smallest distance. We found that on average only 10% of the discharge summary is related to the query. This amount is represented by 1 to 3 paragraphs, compared to an average discharge summary of 10 to 20 paragraphs.

Therefore, including the whole summary to expand the query can potentially result in a noisy query. Once we extract the related paragraphs from the discharge summaries, we remove the stop words and terms with low tf-idf. Then, we expand the original query with the processed text.
4.5 Retrieval Method

We include the topic information by adding $P_{\text{gllda}}(q_i|D, \hat{\theta}_j, \hat{\phi})$ to the standard language model defined as follows\cite{91}:

$$P(q_i|D) = \lambda \left( \frac{N_d}{\mu + N_d} \frac{c(q_i|D)}{|D|} + (1 - \frac{N_d}{\mu + N_d})P(q_i|C) \right) + (1 - \lambda) \left( P_{\text{gllda}}(q_i|D, \hat{\theta}_j, \hat{\phi}) \right)$$

$$P_{\text{gllda}}(q_i|D, \theta_j, \phi) = \sum_{z=1}^{K} p(w|z, \hat{\phi}) p(z|\hat{\theta}_j)$$

(4.4)

where $q_i$ is the vector of query terms, $K$ is the number of topics, $\hat{\phi}$ is the posterior probability estimate of the word mixture for each topic. The value of $\hat{\theta}_j$ is the topic mixture for the document. We set the smoothing parameter $\mu = 1$ and the combining parameter $\lambda = 0.6$. $N_d = J$ is the number of documents in the corpus.

4.6 Results

4.6.1 Experimental Settings

To test the proposed framework, we use the corpus of medical-related documents provided by the Khresmoi project \cite{82}. This collection consists of crawled web pages from health and medicine websites that have been certified by the Health on the Net (HON) Foundation as well as other commonly used healthcare websites. The sites cover a broad range of health topics, and they target both the general public and healthcare professionals. This dataset consists of 1,628,500 documents. Due to the nature of the collection, the number of words in each document ranges from 20 to 2000 words. To extract the most common noun phrases, we annotate 15000 documents (1% of the
corpus) selected randomly. We extract 6883 noun phrases with two or more terms.

We process the corpus in the following manner: First we extract the text content from the document by removing the HTML tags and headers. Then, we remove special, foreign characters and numbers. Subsequently, we replace the noun phrases by a unique identifier, perform stemming and remove stop words.

We use the queries from CLEF eHealth Challenge Task 3 [82], which are formed by 50 medical related queries. In addition to the query, there is a brief description of the user who poses the query (the patient himself, a patient’s family or a nurse). Each of these queries has an associated discharge summary or procedure report from the patient that leads the user to pose the query. These text segments are obtained from the MIMIC II dataset [75] which contains the Electronic Medical Record (EMR) from 33000 patients that entered the ICU.

### 4.6.2 Validation

We test 6 variants of the model for three different number of topics. In three of these variants we use standard bag of words: Topic Model only (TM), Topic Model with Discharge Summaries (TM+DS), and a variant of this method by removing the terms with low tf-idf measure (TM+DSTF) (less than 1). In the other variants, we expand the bag of words models mentioned above with Noun phrases (TM+NPh, TM+NPh+DS, .

<table>
<thead>
<tr>
<th>Feature</th>
<th>Single words</th>
<th>Noun phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary size</td>
<td>98,734</td>
<td>101,497</td>
</tr>
<tr>
<td>Average unique terms per document</td>
<td>767.553</td>
<td>760.522</td>
</tr>
</tbody>
</table>
TM+NPh+DSTF) for $K = 75, 100$ and $150$ topics. Our goal is to test if the number of topics affects the retrieval performance and find the contribution of each model component.

Table 1 shows the statistics of the corpus with bag of words and with noun phrases. Here, we observe that the number of noun phrases does not increase the vocabulary size significantly (around 2.7%).

We compare our method with BM25 with pseudo-relevance feedback as a baseline and with the results of the winner of the CLEF eHealth Challenge Task 3. This approach is based on Markov Random Fields and term expansion using the Medical Subject Headings (MESH) [102]. This method uses ontologies to extract the relationships between query concepts to expand the query. In addition, we compare the performance of our method with Standard Language Models. Our goal is to show the effectiveness of Statistical Topic modeling in incorporating global context in the retrieval task compared to detailed query expansion using query relationships based on ontologies.

### 4.6.3 Experimental Results

Table 2 displays the retrieval results based on Precision at 5 (P@5) and 10 (P@10), Mean Average Precision (MAP) and the Normalized Discounted Cumulative Gain (NDCG). In addition, we show the number of relevant documents retrieved for the 50 queries as a global measure for recall. CLEF eHealth Challenge has a total of 1883 relevant labeled documents. Then, the closer the framework is to this number the better global recall the method has.
We observe that our model is consistently better in the $P@5$, $MAP$ and in the $NDCG$ measures for all the variants tested. Our method outperforms the CLEF eHealth winner by 68% in MAP and by 13% in NDCG. The effectiveness of the Statistical Topic Models in the document retrieval is clearly significant. We observe that the TM variant outperforms the standard Language Model framework by 30% in the $P@10$, 92% in the MAP and 45% in the NDCG measures.

We report that the best performance is achieved when we train the model with 100 topics and when we include the Discharge Summaries. This result shows that discharge summaries provide context to the query and consequently improve the retrieval performance. This improvement is particular evident in ambiguous queries or those which contain acronyms, i.e., Shortness of breath (SOB) vs. crying (sob). In other queries, the addition of the extra context has little or no effect on the performance due to the amount of information already provided by the query.

We find that the use of noun phrases does not improve the overall performance. The main reason is that noun phrases are useful when the user searches for a particular type of disease rather than when he looks for general information.

### 4.7 Discussion

We have presented a method to retrieve relevant healthcare documents using Statistical Topic Modeling methods. We showed that the proposed framework outperformed ontology-based approaches. Despite the richness of knowledge extracted by the
Table 4.2: Mean Performance Results of the base model and the variants of the model for the test set

<table>
<thead>
<tr>
<th>Model</th>
<th>P@5</th>
<th>P@10</th>
<th>MAP</th>
<th>NDCG@10</th>
<th>Doc. Retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (BM25)</td>
<td>0.4520</td>
<td>0.4700</td>
<td>0.3043</td>
<td>0.4169</td>
<td>1651</td>
</tr>
<tr>
<td>CLEF eHealth winner</td>
<td>0.4960</td>
<td>0.5180</td>
<td>0.3108</td>
<td>0.4665</td>
<td>1673</td>
</tr>
<tr>
<td>LM</td>
<td>0.4040</td>
<td>0.4040</td>
<td>0.2666</td>
<td>0.3637</td>
<td>1646</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K=75</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TM</td>
<td>0.5183</td>
<td></td>
<td>0.4828</td>
<td>0.5179</td>
<td>1715</td>
</tr>
<tr>
<td>TM+DS</td>
<td>0.5224</td>
<td></td>
<td>0.4960</td>
<td>0.4998</td>
<td>1014</td>
</tr>
<tr>
<td>TM+DSTF</td>
<td>0.5008</td>
<td></td>
<td>0.5204</td>
<td>0.4958</td>
<td>1023</td>
</tr>
<tr>
<td>TM+NPh</td>
<td>0.4920</td>
<td></td>
<td>0.5040</td>
<td>0.4746</td>
<td>1697</td>
</tr>
<tr>
<td>TM+NPh+DS</td>
<td>0.5060</td>
<td></td>
<td>0.4959</td>
<td>0.5041</td>
<td>1433</td>
</tr>
<tr>
<td>TM+NPh+DSTF</td>
<td>0.5320</td>
<td></td>
<td>0.5102</td>
<td>0.5049</td>
<td>958</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K=100</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TM+NPh+DSTF</td>
<td>0.5224</td>
<td>0.5122</td>
<td>0.4840</td>
<td>0.5117</td>
<td>1722</td>
</tr>
<tr>
<td>TM+NPh+DS</td>
<td>0.5200</td>
<td>0.5000</td>
<td>0.5226</td>
<td>0.5202</td>
<td>1013</td>
</tr>
<tr>
<td>TM+DSTF</td>
<td>0.5160</td>
<td>0.5166</td>
<td>0.4849</td>
<td>0.5088</td>
<td>1694</td>
</tr>
<tr>
<td>TM+NPh+DS</td>
<td>0.5160</td>
<td>0.4645</td>
<td>0.4881</td>
<td>0.4776</td>
<td>991</td>
</tr>
<tr>
<td>TM+NPh+DSTF</td>
<td>0.4020</td>
<td>0.4022</td>
<td>0.4523</td>
<td>0.4235</td>
<td>827</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K=150</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TM+DSTF</td>
<td>0.5265</td>
<td>0.5224</td>
<td>0.4877</td>
<td>0.5186</td>
<td>1718</td>
</tr>
<tr>
<td>TM+NPh+DSTF</td>
<td>0.5326</td>
<td>0.5041</td>
<td>0.5011</td>
<td>0.5155</td>
<td>1508</td>
</tr>
<tr>
<td>TM+NPh+DS</td>
<td>0.5306</td>
<td>0.5163</td>
<td>0.5200</td>
<td>0.5239</td>
<td>1445</td>
</tr>
<tr>
<td>TM+NPh+NPh+DSTF</td>
<td>0.4760</td>
<td>0.5020</td>
<td>0.5167</td>
<td>0.4899</td>
<td>1670</td>
</tr>
<tr>
<td>TM+NPh+DSTF</td>
<td>0.5340</td>
<td>0.4632</td>
<td>0.4762</td>
<td>0.4846</td>
<td>842</td>
</tr>
<tr>
<td>TM+NPh+NPh+DSTF</td>
<td>0.4709</td>
<td>0.4204</td>
<td>0.4464</td>
<td>0.4325</td>
<td>714</td>
</tr>
</tbody>
</table>

ontology-based techniques, Statistical Topic Modeling captured the document context more effectively for a large corpus of documents obtained from multiple sources in an unsupervised form. We found that the incorporation of discharge summaries in the query improved the overall retrieval performance. These summaries are particularly useful for
the case of queries containing general terms and acronyms. Discharge summaries help us to disambiguate query terms by providing additional information to the query.

Further work includes performing a Learning to Rank algorithm to combine the scores of the retrieval of the TM and TM+DS variants having the clarity of the query as a weighting factor.

To effectively incorporate the information contained in the discharged summary in the query, we plan to explore other methods to segment the discharge summaries [19, 87]. In addition, we propose to decompose the paragraphs related to the query into topics. This decomposition would enable us to cluster relevant documents effectively according to their topic mixture. We also plan to exploit the correlations among topics as a method of query disambiguation.
Part II

Dynamic Models and Their Applications in Health State Prediction
Chapter 5

Dynamic Health State Prediction

5.1 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 trigger a medical alarm successfully. 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 to assign them medical resources more efficiently.

The prevailing medical practice relies on frameworks such as the Apache III [46], and SAPS II [44] scores. Both methods, widely used to predict patient mortality in the Intensive Care Unit (ICU), incorporate temporal information in a limited way. The
do this 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 the problem of estimating the patient’s mortality [50, 8, 40, 32]. 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 in time, or dynamic, such as waveforms of vital signs that are discretized into static features. Lee et al. in [50] 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 [8]. 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 [40].

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 high 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 [58]. Previous approaches tackle this problem by segmenting the patient features according to their age group and then calculate the average value for each segment [63, 50]. 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 [47]. 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 [63, 58]. 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 essential information that is potentially useful to predict better the presence of an increase in the probability of mortality [32, 53]. Examples of text include lab reports, admission, doctors and nurse notes.

Ghassemi et. al [32] combine static numerical features such as SAPS II score with topic modeling features from the text of the EMRs. They predict the probability
Table 5.1: Literature Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Dynamic Update</th>
<th>Features</th>
<th>Handles Missing Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache III &amp; SAPS II</td>
<td>No</td>
<td>Static Numerical</td>
<td>No</td>
</tr>
<tr>
<td>Batal 2009</td>
<td>No</td>
<td>Dynamic Numerical</td>
<td>No</td>
</tr>
<tr>
<td>Hug 2009</td>
<td>No</td>
<td>Dynamic Numerical</td>
<td>No</td>
</tr>
<tr>
<td>Lehman 2012</td>
<td>No</td>
<td>Static (Text + Numerical)</td>
<td>No</td>
</tr>
<tr>
<td>Gassemi et al 2014</td>
<td>No</td>
<td>Batch Aggregated Text + Static</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Numerical</td>
<td></td>
</tr>
<tr>
<td>Proposed Method</td>
<td>Yes</td>
<td>Dynamic (Text + Numerical)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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 that 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 5.1 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. We obtain this data from the MIMIC II dataset [75]. 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 5.2 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 7.4. Finally, a discussion of the significance and impact of our proposed model is provided in section 5.7.

5.2 Dynamic Health State Prediction

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.

5.3 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 \( \xi_t \) 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 can 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 particular case of the Generalized Dynamic Linear Models [92]. Here we employ the logit transformation to accommodate our specific context, which leads to the following expressions:

\[
Y_{t,i} \sim Bernoulli(\pi_{t,i}),
\]

\[
\pi_{t,i} = \frac{e^{\xi_{t,i}}}{1 + e^{\xi_{t,i}}},
\]

\[
\xi_{t,i} = \xi_{t-1,i} + \tilde{\theta}_{t,i} + w_{t,i}, \quad w_{t,i} \sim N(0, W_{\xi})
\]

\[
\tilde{\theta}_{t,i} = \lambda \tilde{\theta}_{t-1,i} + \beta X_{t,i} + w_{t,i}, \quad w_{t,i} \sim N(0, W_{\theta})
\]

Here \( \lambda \) is a decay factor that determines the contribution of previous state values in the current one [7]. 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. \( \beta \) represents the vector of regression coefficients we use to combine the observed features. \( W_{\xi} \) and \( W_{\theta} \) are the
evolution variances of $\xi$ 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 $[X_{t+1,i} \ldots X_{t+k,i}]$ is observed. We determine this impact by the following forecast function:

$$\xi_{t+k} = \sum_{r=0}^{k} \lambda^r \tilde{\theta}_t = \tilde{\theta}_t (1 - \lambda^{k+1})/(1 - \lambda) \quad (5.5)$$

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 the 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 can 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:

\[ Y_{t,i} \sim \text{Bernoulli}(\mu_{t,i}), \quad \mu_{t,i} = \frac{e^{F'\theta_{t,i}}}{1 + e^{F'\theta_{t,i}}}, \]

\[ \xi_{t,i} = F'\theta_{t,i} \]

\[ \theta_{t,i} = G\theta_{t-1,i} + [0, \beta X_{t,i}]' + w_t \]

\[ F' = [1, 1] \quad \theta_{t-1,i} = \begin{bmatrix} \xi_{t-1,i} \\ \tilde{\theta}_{t-1,i} \end{bmatrix} \]

\[ G = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} \quad w_t \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} W_\xi & 0 \\ 0 & W_\theta \end{bmatrix} \right) \]

Figure 5.1 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 \( \Phi \) defined as:

\[ \Phi = \lambda, W_\theta, W_\xi, \beta. \]

This representation is flexible enough to expand the model and to incorporate different weighting vectors \( \beta \) for various 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 can 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 5 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.

### 5.4 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.
Algorithm 5 Proposed Method

Extract text and topic Features

Construct time series

Impute Missing Values using EM based method

Use an initial guess of the parameters \( \Phi = (\lambda, W_\xi, W_\theta, \beta) \)

Define \( P \) to be the number of patients in the dataset

repeat

\( \text{for } i \leftarrow 1 \text{ to } P \text{ do} \)

Define \( T_i \) to be the number of time steps of the series for a patient \( i \)

\( \text{for } t \leftarrow 1 \text{ to } T_i \text{ do} \)

Estimate the value of \( m_{t,i} \) and \( C_{t,i} \) using Filtering Equations described in subsection 5.4.1

end for

\( \text{for } t \leftarrow T_i \text{ to } 1 \text{ do} \)

Estimate the expected value \( s_{t,i} = E(\theta_{t,i}|D_{1:T_i}) \) and variance \( S_{t,i} = \text{var}(\theta_{t,i}|D_{1:T_i}) \) of the hidden states using FFBS algorithm described in subsection 5.4.2

end for

end for

Estimate the parameters \( \Phi \) that maximize the likelihood \( P(\theta, \Phi|D_{1:T_i,1:P}) \)

until Convergence
5.4.1 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 \( t = 1, \ldots, T \) 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 \( \Phi \). Here, the observations are assumed to be continuous and normally distributed as defined in the standard DLM [92, 73]. 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 [29].

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

\[
\ell(\theta_{t,i}|D_{1:t,i}, \Phi) \approx -\frac{1}{2}(\theta_{t,i} - a_{t,i})'R_{t,i}^{-1}(\theta_{t,i} - a_{t,i})
- \ln(1 + \exp(-Y_{t,i}F'\theta_{t,i}))
\]

(5.8)

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 \( \phi \). 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:
\[ a_{t,i} = Gm_{t-1,i} + \beta X_{t,i} \] 
\[ R_{t,i} = GC_{t,i}G' + W \]

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:

\[ \theta_{t,i} | D_{1:t,i}, \Phi \sim N(m_{t,i}, C_{t,i}), \]
\[ m_{t,i} = \theta_{t,i}^{MAP}, \quad C_{t,i} = -H^{-1}(\theta_{t,i}^{MAP}) \]
\[ \theta_{t,i}^{MAP} = \arg \max_{\theta_{t,i}} \ell(\theta_{t,i} | D_{1:t,i}, \Phi) \]  
\[ \frac{\partial \ell(\theta_{t,i})}{\partial \theta_{t,i}} = \frac{FY_{t,i}}{1 + \exp\{Y_{t,i}F'\theta_{t,i}\}} - R_{t,i}^{-1}(\theta_{t,i} - a_{t,i}), \]
\[ \frac{\partial^2 \ell(\theta_{t,i})}{\partial \theta_{t,i}^2} = \frac{-FF'}{(1 + \exp\{-Y_{t,i}F'\theta_{t,i}\})(1 + \exp\{Y_{t,i}F\theta_{t,i}\})} - R_{t,i}^{-1} \]

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:

\[ \theta_{t,i} | D_{1:t,i} \sim N(m_{t,i}, C_{t,i}), \quad m_{t,i} = \theta_{t,i}^{MAP}, \]
\[ C_{t,i}^{-1} = \frac{FF'}{(1 + \exp\{-Y_{t,i}F'\theta_{t,i}^{MAP}\})(1 + \exp\{Y_{t,i}F\theta_{t,i}^{MAP}\})} + R_{t,i}^{-1} \]

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.

### 5.4.2 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 \( \Phi \) [73, 31]. 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 5.4.1. Then, we obtain the smoothing mean $s_{T,i}$ and variance $S_{T,i}$ of the state variables at time $T$, $\theta_{T,i}|\theta_{D_{1:T,i}} \sim N(s_{T,i}, S_{T,i})$. 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 an entirely dynamic model with feedback. One variant of the model is a dynamic model with open loop feedback (no feedback about the future). We fit this model by using forward filtering (FF) using the filtering equations of subsection 5.4.1. For numerical stability, we use the singular value decomposition (SVD) based approach detailed in [101] to find the values of $s_{t,i}$ and $S_{t,i}$. We describe the steps needed to perform the FFBS in our model in Algorithm 6.

### 5.4.3 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 determine the expected state values using the Forward Filtering Backward Smoothing (FFBS) algorithm, explained in the previous subsection. In the M-step, we estimate the parameters that maximize the likelihood function estimated in the E-step.

The likelihood function is defined as follows:
Algorithm 6 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 5.4.2.

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)

$$s_{t,i} = m_{t,i} + C_{t,i}G'R_{t+1,i}^{-1}(s_{t+1,i} - a_{t+1,i})$$

$$S_{t,i} = C_{t,i} - C_{t,i}G'R_{t+1,i}^{-1}(R_{t+1,i} + S_{t+1,i})R_{t+1,i}^{-1}GC_{t,i}$$

$$R_{t+1,i} = GC_{t,i}G' + W_{t,i}$$

$$a_{t+1,i} = Gm_{t,i} + \beta X_{t+1,i}$$

end for
\[
p(\theta, \Phi | D_{1:T}) = \prod_{i=1}^{P} p(\theta_{1,i}) \prod_{t=2}^{T_i} p(\theta_{t,i} | \theta_{t-1,i}) p(\theta_{t,i} | \phi, D_{T_i})
\] (5.13)

The log-likelihood is concave given \( G, \lambda, W_{\theta} \) and \( W_{\xi} \). 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 [31] and after some algebra, we obtain the M-step update expressions given the current value of parameters as follows:

\[
G_{new} = \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t-1,i} | \theta_{t,i}'] \right]^{-1} \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t-1,i} | \theta_{t-1,i}'] \right]^{-1}
\]

\[
E[\theta_{t-1,i} | \theta_{t,i}'] = m_{t-1,i} s_{t,i}' + L_{t-1,i} (S_{t,i} + (s_{t,i} - a_{t,i}) s_{t,i}') - \beta X_{t,i} s_{t,i}'
\] (5.14)

\[
E[\theta_{t-1,i} | \theta_{t-1,i}'] = S_{t-1,i} + s_{t-1,i} s_{t-1,i}'
\]

\[
W_{new} = \frac{1}{\sum_{i=1}^{P} \sum_{t=1}^{T_i}} \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t,i} | \theta_{t,i}'] - E[(G\theta_{t-1,i} + \beta X_{t,i}) \theta_{t,i}']
\]

\[
E[\theta_{t,i} | \theta_{t,i}'] = S_{t,i} + s_{t,i} s_{t,i}'
\]

\[
E[G\theta_{t-1,i} | \theta_{t,i}'] = G \left[ m_{t-1,i} s_{t,i}' + L_{t-1,i} (S_{t,i} + (s_{t,i} - a_{t,i}) s_{t,i}') \right]
\]

\[
E[\beta X_{t,i} | \theta_{t,i}'] = \beta X_{t,i} s_{t,i}'
\]

\[
\beta_{new} = \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} (s_{t,i} - G s_{t-1,i}) X_{t,i} \right] \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} X_{t,i}' X_{t,i} \right]^{-1}
\] (5.16)
where:

\begin{align*}
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{align*}

(5.17)

\[ \lambda_{\text{new}}^\text{new} = G_{2,2}^\text{new}, \quad W_\xi^\text{new} = W_{1,1}^\text{new}, \quad W_\xi^\text{new} = W_{2,2}^\text{new}. \]

\( m_{t,i} \) and \( C_{t,i} \) for \( t = 1 \ldots T_i \) are calculated during the time series filtering step described in subsection 5.4.1. Given these estimations, backward smoothing is performed, based on standard Kalman smoothing recursion described in subsection 5.4.2, 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.

### 5.4.4 Text Processing

Standard approaches to estimating the probability of mortality, Apache III, and SAPS II scores, do not incorporate text information. One of the main challenges researchers face is to combine 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 can 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 to extract different features that are later integrated into the model. Algorithm 7 shows a summary of the feature extraction process. Figure 5.2 depicts the text-based feature representation.

5.4.4.1 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 that are later integrated into the dynamic model.

We extract relevant noun phrases by annotating the discharge summaries not included in the training data using the Clinical Text Analysis and Knowledge Extraction System (cTAKES)[76] and Metamap [5]. These Natural Language processing tools for the medical domain extract clinically 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 for the patient is discharged from the ICU.

After extracting annotating and extracting all the noun phrases, we select only
those which describe a disease, procedure or medication using the medical ontologies provided by SNOMED [80]. 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 smaller medical noun phrases. Therefore, we decouple these phrases into their smallest possible unit. Through using tf-idf term selection, we select the most important noun phrases and remove those with a 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 obtain two types of features: term and topic based features, which we describe below.

5.4.5 Term Based Features

We incorporate into the model a term-based feature using the derived noun phrases and terms from the EMR text 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 a good predictive performance, and it is computationally feasible for this problem [27].

To make this classification possible, we reduce the large vocabulary size of the corpus by extracting the most discriminative terms using $\chi^2$ test [26]. We performed this analysis on the whole corpus. Our goal is to obtain a global estimate from the
entire corpus to reduce the bias resulting from the term selection.

### 5.4.5.1 Topic Based Features

The second set of text features is based on Statistical Topic Modeling. These models allow us to minimize the dimensionality of the term space to a smaller feature space of latent ”topics”. In addition, we can 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 described 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 can improve the performance of the probability of mortality estimation.

For this application, we fit a GD-LDA model developed by authors of [16] to extract the topics from the corpus set. This data consists of all the processed text entry
notes (noun phrases + terms) of all the patients. GDLDA, which is a generalization of LDA [11], 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. This 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) [12].

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 describe as common words those that do not have healthcare related information by comparing them with the ontologies from the UMLS using MetaMap[5]. These ontologies provide information about health care treatments, drugs, and diseases.

After removing the background topics, we select 10 most discriminative topics using the $\chi^2$ 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.
5.5 Missing Features Estimation

The proposed framework mentioned above assumes that we observe all the patient’s features at each point in time. When feature values are not found, 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 require particular labs 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 employing a Regularized Expectation Maximization method [77]. 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:

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

where \( \hat{\mu}_m \) is the mean estimate of the missing features of a record and \( \mu_a \) is the average estimate of 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 \( \gamma \) 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 [77].
Algorithm 7 Feature Extraction

Numerical Feature Extraction:

Transform features using Apache Score III weights
Perform $\chi^2$ test on all the observed features
Select the features with higher separation score

Term based Feature Extraction:

Construct term frequency matrix
Estimate the $\chi^2$ 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

5.6 Estimation of Probability of Mortality

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.
5.6.1 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 efficiently and to aid doctors and nurses ahead of time. For the situation that 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 [75]. 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. To compare our approach with other methods from the literature, we only study the adult patients (over 18 years of age). In our method, we do not exclude any patient due to a particular 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 we incorporate into the model are selected using the $\chi^2$ 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 [46, 44]. 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
methods 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 low blood pressure as to have it to be 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 [46].

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 5.5.
Table 5.2: Performance of the 3 variations of our model, 3 different methods used in the literature and 2 static classification methods

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

\(^1\)These are the results reported in [32]. Authors report results without cross validation. In addition authors remove patients with less than 100 stop words
We construct a time series sequence for each of the 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 about 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 standard practices to predict the probability of mortality such as the Apache Score [46] and SAPS II [44] 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 is 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 [16] model using all the text entries and $K = [50, 75, 100]$ topics. Figure 5.3 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 (collecting 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 [32]. The resulting topics provide a better context, which aids to improve 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 employed in the literature: Apache III Score [46], SAPS II [44] and the results provided by Ghassemi et al. [32].
5.6.2 Experimental Results

We fit the parameters of our proposed model using the Expectation Maximization approach described in section 5.4 using 50 iterations for the 3 sets of features. We observe that the average estimated value of $\lambda$ is around 0.02 for numerical features only and 0.21 for text and topic features. These values of $\lambda$ 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. We also display the performance of 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 [32] which are calculated 24 hours after the
patient is admitted to the ICU.

Table 5.2 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. To compare our results with methods in the literature, we report findings in those time stamps. However, our method can 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 Random Forest (0.7985 vs. 0.7460). We note that our model 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 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). Also, our method can 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 curve. We observe that all three variants of our method have better performance consistently 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 5.3 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 Random Forests). Therefore, our method has a good predictive performance for the genuine 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 useful in estimating the probability of mortality correctly.

We also compare the performance of our method against Support Vector Ma-
Table 5.3: Performance based F Scores of our model and related methods of the literature in $t = 24, 48$ and $72$ hours after patient admission

<table>
<thead>
<tr>
<th>Method</th>
<th>24 hours</th>
<th>48 hours</th>
<th>72 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache III</td>
<td>0.4662</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SAPS II</td>
<td>0.3863</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.4137</td>
<td>0.4169</td>
<td>0.3975</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.3929</td>
<td>0.4406</td>
<td>0.3977</td>
</tr>
<tr>
<td>Proposed Method (only numerical features)</td>
<td>0.4629</td>
<td>0.5409</td>
<td>0.4570</td>
</tr>
<tr>
<td>Proposed Method (numerical + term based)</td>
<td>0.4929</td>
<td>0.5806</td>
<td>0.4853</td>
</tr>
<tr>
<td>Proposed Method (numerical + term + topic features)</td>
<td>0.5229</td>
<td>0.5450</td>
<td>0.5367</td>
</tr>
</tbody>
</table>

Table 5.4: Progression of the average probability of mortality in $t=24,48,72$ hours as predicted by our model with 3 different sets of features

<table>
<thead>
<tr>
<th>Patients</th>
<th>24 Hours</th>
<th>48 Hours</th>
<th>72 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical Information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovered</td>
<td>0.8321</td>
<td>0.8420</td>
<td>0.8420</td>
</tr>
<tr>
<td>Died</td>
<td>0.9500</td>
<td>0.9621</td>
<td>0.9620</td>
</tr>
<tr>
<td>Numerical Information + Text Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovered</td>
<td>0.8303</td>
<td>0.8395</td>
<td>0.8404</td>
</tr>
<tr>
<td>Died</td>
<td>0.9501</td>
<td>0.9695</td>
<td>0.9685</td>
</tr>
<tr>
<td>Numerical Information + Topic Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recovered</td>
<td>0.8303</td>
<td>0.8321</td>
<td>0.8414</td>
</tr>
<tr>
<td>Died</td>
<td>0.9541</td>
<td>0.9685</td>
<td>0.9621</td>
</tr>
</tbody>
</table>

chines (SVM). However, the results of this method are highly dependent on the imputation model used. Performing single value imputation with the missing data leads to very poor results that are not comparable with the other methods are shown in Table 5.2. Thus, we do not report the results of this technique.

Table 5.4 shows the progression of the probability of mortality of patients who died and patients that recovered and exited the ICU using the 3 variants of our model. Here we observe an apparent difference in the probability of mortality between both groups of patients. Figure 5.6.2 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
Figure 5.4: 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.

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 can predict this fact accurately 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. Also, we plan to model the probability of reentry to the ICU, as this is another perspective measure of effective patient care.

5.6.3 Computational Complexity

Our model has a continuous state space compared to the discrete space of other models such as Hidden Markov Models (HMM). This fact implies that the quality of our prediction measure is superior to the discrete approximation of HMMs with less computational complexity. The complexity of the proposed approach is: \( O(P \times N^2 \times T) \) 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: $O(P \times N^K \times T)$. Here, $N$ 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.

5.6.4 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 [32]. 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 illnesses.

Our method obtains significantly better specificity performance (percentage of true negatives detected) than other methods. A low specificity value implies the existence of a vast 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 an actual 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 critical in the correct detection of true alarms. Correctly detecting all real alarms is desirable since the cost of not identifying a patient who is very ill and dies is very high.
5.7 Discussion

In this chapter, 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 provides 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 can 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.

We find that by using static methods, which are currently used in the literature and real applications, they oversee relevant patient behavior. We can only achieve this by taking into account the evolution of the patient, as it is performed by physicians in a real scenario. In this context, physicians take into account previous patient readings together with the current ones to make decisions about the patient.

Besides, by waiting 24 hours before performing APACHE score and other related scores, we are only targeting patients with long recovery periods. In reality, our data shows that 35% of the adult patients admitted to the ICU are there 24 hours or
less. Therefore, the current practice in prediction models do not provide an adequate prediction for short-term stay patients, since the data used to predict these models is biased toward long recovery time patients. Therefore, it is important to provide short and long-term prediction methods to capture both patient types.

Text information gives a context that numerical features cannot provide. The use of these features has been shown to increase the performance of the mortality prediction [32].

We have found that the although the admission reports provide a very comprehensive summary of the initial state of the patient, they need to be further segmented and processed. This extra processing would help us to provide better estimates, which would be used in the model. The lab reports and the doctor notes tend to be more concise and carry more sequential information. Further extension of the work will include to determine which other text features we should incorporate into the model to improve further the performance of the method.

Also, we plan to incorporate previous medical knowledge into the model using ontologies and to extract the main concepts and relationships between different patient’s symptoms to predict better 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 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 that 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. Our final goal is to provide reliable decision support tools that aid the physicians to process all the patient reading from different sources effectively in an automated form.
Chapter 6

Dynamic Estimation of the Probability of Patient Readmission to the ICU using Electronic Medical Records

6.1 Introduction

Currently, the accurate and opportune prediction of patient readmission to the ICU shortly after he is transferred to a lower level care is of great interest to health providers. Tools, which estimate the probability of patient relapse and readmission to the ICU, aid physicians and healthcare providers to determine the possible resources that should allocate to the patient. They also help to discover the potential causes of relapse that could lead the patient to be readmitted to the ICU. The timely and accurate estimation of patient’s probability of readmission allows us to trigger a medical alarm successfully before the patient is transferred from the ICU.
This probability estimation also permits the early identification of patients with elevated risk of readmission. As a result, health care providers can differentiate those patients from the ones who are stable and less likely to return and assign medical resources more effectively.

Most of the existent methods in the literature[28, 72] rely on the use of static classifiers that do not take into account the evolution of the patient nor the dynamic nature of the patient’s features. To overcome these challenges, we propose a dynamic method based on Bayesian Time Series and Dynamic Linear Models (DLM) to estimate the probability of readmission before the patient is transferred to another health unit.

We want to indicate the existence of a possible medical alarm. Our contribution is summarized as follows: We model the probability of patient readmission as an aggregated latent state, which is updated each time new features are observed (lab results, vital signals, etc.). We feed our model with heterogeneous data obtained from his Electronic Medical Records (EMRs) which consists of text and numerical data with both discrete and continuous variables. We incorporate the text information into the model by developing a method that converts the unstructured text information into discriminative features; which are later integrated into the model. Finally, we address the missing values problem by estimating those values using a Regularized Expectation Maximization (EM) based method.

In this context, we find that the dynamic estimation of the probability of patient readmission to the ICU provides a more accurate prediction when compared with other methods in the literature and other static prediction methods.
6.2 Related Work

The prevailing medical practice relies on frameworks such as the Apache III [46], and SAPS II [44] scores. Both methods, widely used to predict patient mortality in the Intensive Care Unit (ICU), are used as proxy to estimate the likelihood that a patient is not ready to be transferred to a lower level care. Therefore, there is a high probability of being readmitted if he is discharged from the ICU. These methods incorporate the 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 (not readiness to be transferred from the ICU). Moreover, these scores are only estimated once during the entire stay in the ICU, which may not indicate whether the patient would recover and be successfully transferred from the ICU in the future.

Data Mining has been previously used to address the problem of estimating the likelihood that a patient readmission to the ICU [28, 72]. Other prediction methods, such as Batal et al.[8], 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. However, this model does not take into account the evolution of the patient in time. Most of the existing solutions rely on training a static classifier with a patient’s observed feature vector. These features are mainly static, such as lab reports produced at a given point in time or the estimated APACHE score at the discharge time.
Furthermore, most of the methods mentioned above assume the availability of all the features at the prediction 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 high volume of missing data because not all the features are collected (lab results, vital signal, 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 [58]. Previous approaches tackle this problem by segmenting the patient features according to their age group and then calculate the average value for each segment [63, 50]. 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 [47]. 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 [63, 58]. Overall, these methods do not take into account the temporal aspect of the missing data where current features values 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 critical information that is potentially useful to predict better the likelihood that a patient is readmitted to the ICU. Examples of text include lab reports, admission, doctors and nurse notes. Ghassemi et. al [32] combine static numerical features, such as SAPS II score, with topic modeling features from the text of the EMRs to estimate the probability that a patient die after 30 days of being discharged. They achieve this by using Support
Vector Machines (SVM).

Our proposed approach combines text and numerical information in a dynamic setting that allows us to predict the patient readmission before the patient is transferred from the ICU to other hospital areas. In addition, our proposed approach takes into account the evolution of the patient health state and temporal aspect of the patient features to predict how likely he will relapse and will be readmitted to the ICU.

6.3 Definition of Probability of Readmission as a Latent State

We define the that a patient \( i \) would be readmitted to the ICU in the next 30 days \( Y_{t,i} \) or not \( Y_{t,i} = -1 \) if it is discharged from the ICU at time at time \( t \) as a binary variable. \( Y_{t,i} \) is 1 with a probability of \( \pi_{t,i} \), which represents the probability that this patient is readmitted if he is transferred from the ICU at time \( t \). (probability of readmission). This probability is a function of a latent state \( \theta_{t,i} = [\xi_{t,i}, \tilde{\theta}_{t,i}] \). This latent state is formed by the estimation of the log-odds of the probability of readmission in previous steps \( \xi_{t-1,i} \) and the patient features observed at time \( t \), \( \tilde{\theta}_{t,i} \). The value of \( \tilde{\theta}_{t,i} \) is obtained by combining a set of observed features \( X_{t,i} \) obtained from the EMR at time \( t \) and the value of this combination at previous steps, \( \tilde{\theta}_{t-1,i} \).

In this framework, we are able to include both the patient’s features and his health context obtained from previous time steps, which is not accounted for in the static classification frameworks. Our proposed model is a particular case of the Generalized
Dynamic Linear Models (GDLM)[92]. Here, we employ the logistic transformation to accommodate our specific context. This leads to the following expressions:

\[ Y_{t,i} \sim \text{Bernoulli}(\pi_{t,i}), \quad \pi_{t,i} = \frac{e^{\xi_{t,i}}}{1 + e^{\xi_{t,i}}}, \quad (6.1) \]

\[ \xi_{t,i} = \xi_{t-1,i} + \tilde{\theta}_{t,i} + w_{t,i}^\xi, \quad w_{t,i}^\xi \sim N(0, W_\xi) \quad (6.2) \]

\[ \tilde{\theta}_{t,i} = \lambda\tilde{\theta}_{t-1,i} + \beta X_{t,i} + w_{t,i}^\theta, \quad w_{t,i}^\theta \sim N(0, W_\theta) \quad (6.3) \]

Here \( \lambda \) is a decay factor that determines the contribution of previous feature values in the current value of \( \tilde{\theta}_{t,i} \). The vector \( X_{t,i} \) is constructed from the patient’s observed lab test results, vital signals readings, text notes features, and demographics (features). In this model, we assume that most of the values of \( X_{t,i} \) are observed. In later subsections of this paper, we explain how we model and estimate the missing values of the feature vector. The vector \( \beta \) represents the regression coefficients we use to combine the observed features. The value of \( \tilde{\theta}_{t,i} \) can take both positive and negative values. Thus, we can increase or decrease the probability of readmission using the observed features \( X_{t,i} \). \( W_\xi \) and \( W_\theta \) are the evolution variances of \( \xi \) and \( \tilde{\theta} \) respectively.

The value of \( \xi_t \) reflects the log-odds effect on the probability of readmission \( \pi_{t,i} \) by previously observed features contained in the state \( \theta_{t,i} \). To illustrate this effect, 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 values of \([X_{t+1,i}, \ldots, X_{t+k,i}]\) are observed. This impact is determined by the following forecast function:
\[ \xi_{t+k} = \sum_{r=0}^{k} \lambda^r \tilde{\theta}_t = \tilde{\theta}_t (1 - \lambda^{k+1}) / (1 - \lambda) \]  

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 the patient probability of readmission even when no measurements are available at a given time \( t + k \). In addition, the effect does not decrease over time, as opposed to \( \tilde{\theta}_t \) (observed features). Each time there are new observations available, the value of the effect \( \xi_{t,i} \) is updated using equation 6.4.

### 6.4 Model Fitting

Figure 5.1 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 \( \Phi \) defined as: \( \Phi = \lambda, W_\theta, W_\xi, \beta \). This representation is flexible enough to expand the model and to incorporate different weighting vectors \( \beta \) for different patient groups with a particular disease or age range.

We fit the model using Dynamic Linear Models with Logistic Transformation. This model incorporates the user features into an aggregated patient state that evolves over time, in contrast to static classification models. Also, we train the model using the entire patient’s stay path as opposed to individual time steps. By performing this, we take into account the uncertainty about the future in the estimation of the probability.
Our proposed approach allows us to predict future values of the state as more readings become available. Consequently, we can dynamically estimate the current patient probability of readmission and predict its evolution using the predictive forecast function of the latent state. Figure 6.1 describes the fitting steps for the proposed model.

We train the model using an iterative method based on Expectation Maximization (EM). This method consists of 2 stages: E and M steps. In the E step, we estimate the latent state of the patient $i$, $\theta_{t,i}$ using the Forward Filtering Backward Smoothing method (FFBS)[73]. In this method, we estimate first the latent state $\theta_{t,i}$ using the values of the observed features $X_{t,i}$ and the state value of the previous time step $\theta_{t-1,i}$ (Forward Filtering). Once we have estimated the entire path, we correct the estimated latent state backward using the estimated state values of future time steps.
steps $\theta_{t+1,i}$ (Backward Smoothing). By combining the Forward Filtering (FF) and the Backward Smoothing (BS), we guarantee the construction of an entirely dynamic model with feedback. Here, previous values of the path affect the current latent state while accounting for the future uncertainty. One variant of this dynamic model is to train the model with no feedback about the future (open loop feedback). To achieve this, we fit the model using the latent states obtained with the Forward Filtering (FF) step only.

The M-step consists of estimating the values of the parameters $\Phi$ that optimize the latent patient paths (probability of readmission path) previously estimated in the E-step. Then, we repeat the E and the M steps until convergence.

We predict the probability of readmission in the test partition assuming that we do not know the outcome of the patient. Therefore, we estimate the latent state $\theta_{t,i}$ using the estimated parameters and previous values of the latent state $\theta_{t-1,i}$ using Forward Filtering.

Algorithm 7 shows the method used to extract the numerical and text features from the events described in the EMR.

### 6.5 Experimental Settings and Validation

We test our approach by estimating the probability that a patient discharged from the ICU and transferred to other areas of the hospital would be readmitted in the near future. This prediction is critically important in the accurate prognosis of patient health state (we want to avoid health complications in the patient by discharging him
before time). Besides, this prediction is critical to assign medical resources effectively.

For the situation that we are modeling, we predict the probability that the patient would be readmitted if he is discharged at time $t$ using the information available in his EMR before he is discharged. The EMRs are obtained from the MIMIC II data set [75]. This dataset contains text and numerical information that describes procedures, medications and vital signs readings for 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 of hospitals from Boston Area.

We validate our method using the EMRs of 15,000 patients selected randomly. To compare our approach with other methods from the literature, we only study the adult patients (over 18 years of age). We do not exclude any patient due to a specific illness; this data consists of 11,648 people. We report our results using five-fold cross validation, taking 80% of the patients as training set and the remaining 20% as test set. We construct the time series of each patient by aggregating the patient information every 3 hours.

We report the probability of patient readmission after $t = 24$ and 48 hours and at the time of discharge. Our goal is to test the prediction capability of the proposed approach at different time steps. We compare our method with related methods of the literature used as a proxy to estimate the probability of readmission such as Apache Score [46] and SAPS II [44]. In addition to these methods, we compare the proposed approach with static classification methods such as Naive Bayes and Random Forests.

\[ ^1 \text{K-fold cross-validation provides a more robust prediction evaluation than leave-one-patient-out-method because K-fold cross validation uses less training data} \]
We train Random Forest with 50 trees with a sliding window for $t = 24, 48$ and before patient discharge as in our proposed method.

6.6 Results

When analyzing the time series of the patients, we detect that the 57% of the patients stayed 24 hours of less. Thus standard practices used as a proxy to identify the probability of a patient readmission such as Apache Score and SAPS II are not applicable for those patients. We also detect that after dividing the features into quasi-static and dynamic features 34% of the features are not observed. Therefore, the method to fill out missing values would have a high impact on the estimation of the results.

Figure 5.3 shows the obtained topics after constructing noun phrases and removing background topics. Note that by performing these steps we are able to get cleaner topics, each one related to a particular disease describing symptoms and body parts than those provided by Ghassemi et al. [32]. In addition to the quality of the topics, we also evaluate the quality of the topic-based classification output using the document topic mixture as features. Here we observe that 72% of the notes from the readmitted patients and 52% the not readmitted patients were classified correctly using topic-based features. In addition, we evaluate the performance of the classification using term-based features. Here we observe that 53% of the notes from readmitted patients were classified correctly and 10% of the notes from patients not readmitted were classified correctly. When comparing both features, we observe that topic-based
classification is more accurate than term-based classification. In this context, we corroborate our hypothesis that topics capture the global context of the text notes and improve the classification performance when compared with a term-based classification.

Table 5.2 shows the AUC, sensitivity and specificity for the proposed method and other methods proposed in the literature. Here we observe that our proposed framework outperforms the other methods in the three reported measures and the time steps reported. In the other methods, we notice that their prediction performance decreases after 24 hours. Compared to these methods, our proposed approach does not reduce the performance significantly after 24 hours of patient admission. The reason behind this fact is that the evolution of the latent state and the compilation of the feature data from previous time steps into the current one.

In addition, our proposed approach has a good balance between sensitivity and specificity. A high specificity value implies the existence of a small number of false alarms. In a real scenario, this measure has a great 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. But at the same time, they need to be predicting the existence of an actual alarm accurately. Correctly detecting all real alarms is desirable since the cost of not identifying a patient who is not ready to be discharged who is very ill and dies outside the ICU is very high.

In table 6.1, we also observe that mortality prediction models such as Apache III and SAPS II provide a good proxy to estimate the probability of readmission. However, the main limitation of these methods is that they tend to overestimate the prob-
ability of readmission as we can observe in the specificity measure. The best static prediction method is based on Random Forests at $t = 24$ hours.

Note that the combination of topic, text based and numerical features provide the best prediction performance. Therefore, we corroborate our hypothesis that text data provides complementary information to numerical features and increases the performance in the prediction of patient readmission. When analyzing the average probability of readmission using our model with all the features, we notice that this probability increases from 0.69 in $t = 24$ hours to 0.89 before patient discharge to those who were not readmitted. On the other hand, this probability of mortality change from 0.90 in $t = 24$ hours to 0.92 before the patient is discharged in those who were readmitted to the ICU. This shows that the stay length affects the estimation of the probability of patient readmission. This phenomenon can also explain the performance decrease of the other prediction methods.
Table 6.1: Performance of the 3 variations of our model, 2 different methods used in the literature and 2 static classification methods in terms of Sensitivity, Specificity and AUC

<table>
<thead>
<tr>
<th>Model Type</th>
<th>24 hours</th>
<th>48 hours</th>
<th>Before Patient Discharge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache III</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.9258</td>
<td>0.4042</td>
<td>0.8665</td>
</tr>
<tr>
<td>Spec.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAPS II</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.8707</td>
<td>0.4280</td>
<td>0.8119</td>
</tr>
<tr>
<td>Spec.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.8812</td>
<td>0.1289</td>
<td>0.5320</td>
</tr>
<tr>
<td>Spec.</td>
<td>0.8465</td>
<td>0.1286</td>
<td>0.5916</td>
</tr>
<tr>
<td>AUC</td>
<td>0.8355</td>
<td>10.66</td>
<td>0.5690</td>
</tr>
<tr>
<td>Random Forests</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.9294</td>
<td>0.8006</td>
<td>0.8574</td>
</tr>
<tr>
<td>Spec.</td>
<td>0.928</td>
<td>0.2247</td>
<td>0.7587</td>
</tr>
<tr>
<td>AUC</td>
<td>0.9016</td>
<td>0.2435</td>
<td>0.7263</td>
</tr>
<tr>
<td>Proposed method Numerical Features</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.8094</td>
<td>0.8116</td>
<td>0.8110</td>
</tr>
<tr>
<td>Spec.</td>
<td>0.8870</td>
<td>0.5069</td>
<td>0.8094</td>
</tr>
<tr>
<td>AUC</td>
<td>0.8678</td>
<td>0.6500</td>
<td>0.8071</td>
</tr>
<tr>
<td>Proposed method Text and Numerical Features</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.8692</td>
<td>0.9244</td>
<td>0.9070</td>
</tr>
<tr>
<td>Spec.</td>
<td>0.9087</td>
<td>0.8494</td>
<td>0.8429</td>
</tr>
<tr>
<td>AUC</td>
<td>0.8984</td>
<td>0.8445</td>
<td>0.8202</td>
</tr>
<tr>
<td>Proposed method Topic, Text and Numerical Features</td>
<td>Sens.</td>
<td>Spec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Sens.</td>
<td>0.9043</td>
<td>0.8833</td>
<td>0.9289</td>
</tr>
<tr>
<td>Spec.</td>
<td>0.9138</td>
<td>0.9378</td>
<td>0.9412</td>
</tr>
<tr>
<td>AUC</td>
<td>0.9149</td>
<td>0.8964</td>
<td>0.9274</td>
</tr>
</tbody>
</table>
6.7 Discussion

In this chapter, we have proposed a dynamic model that combines heterogeneous data from the patient Electronic Medical Records to predict the patient readmission to the ICU. The accurate prediction of patient readmission would aid the physicians and medical staff to perform a good allocation of medical resources. The efficient allocation of resource could potentially reduce the cost of patient care. In addition, the reduction of false alarms in the prediction would allow physicians to be available when an actual alarm arrives.

We have shown that methods that use the prediction of the probability of mortality as a proxy for the probability of patient readmission tend to overestimate this probability (high sensitivity and low specificity). The main reason behind this fact is that methods used in the current practice tend to overestimate the probability of mortality. This fact affects the performance in prediction the probability of readmission.

Our proposed model targets to estimate the probability of readmission as opposed to using another measure as a proxy, which is an important aspect to evaluate when we fit the model. Patients that are readmitted in the ICU may have a more diverse profile in observations than the ones with a high probability of mortality, such as chronic conditions that make the patient be readmitted in the ICU.

The use of an aggregated patient state, which combines current features with previously observed ones, allows us to predict the probability of readmission even if not new features are observed. The results of the proposed model depend on the quality
and quantity of the patient observed features. The more features values are observed for a given patient in his EMR and the more diverse patient pool data is, the more accurate readmission prediction would be.

Future path in this direction would integrate treatments and medications to the latent state to specify how those affect the evolution of the patient. To integrate this, we would need to segment the medications according to their nature and doses.

The current model provides a single aggregated latent state. Future work includes expanding the model to create different latent states, one latent state for each body subsystem. Our final goal is to obtain a global prediction estimate that results from the combination of the subsystems latent states. In the same manner, we plan to expand the model to accommodate different feature weight vectors, one for each disease group. This is especially important for the cardiac diseases, which behavior could be significantly different than other types of illnesses.
Chapter 7

Subsystem Failure Prediction

In this chapter, we expand the model described in previous chapters. We describe the method to construct the probability of subsystem failure as a latent state and the framework we use to estimate the probability of mortality. In addition, we outline the methodology to extract numerical and discriminative text features and the method to fill out missing values, which later will be integrated into the model.

7.1 Modeling the Probability of Subsystem Failure as an evolving Latent State

We define a patient $i$ to have a set of subsystems $j = 1 \ldots J$. Each subsystem $j$ can fail $Y_{t,i}^{(j)} = 1$ or not $Y_{t,i}^{(j)} = -1$ at time $t$. We model this output as a binary variable. Therefore $Y_{t,i}^{(j)}$ has a Bernoulli distribution with probability of $\pi_{t,i}^{(j)}$, which we define as the propensity of the system $j$ to fail for the patient $i$ at time $t$. This probability is a
function of a latent state \( \theta_{t,i}^{(j)} \) at time \( t \). The value of \( \tilde{\theta}_{t,i}^{(j)} \) is calculated by combining a set of observed features \( X_{t,i}^{(j)} \) (measurements and procedures) pertaining for each particular subsystem and the value of the latent state at a previous time step \( t-1, \tilde{\theta}_{t-1,i} \).

In this framework, we can include the dynamic change in both features and health context from previous observations. These dynamics are not accounted for when using static classification and some of the state transition based frameworks such Hidden Markov Models. Our proposed model is a special case of the Generalized Dynamic Linear Models [92]. Here we employ the logit transformation to accommodate our specific context. This leads to the following expressions:

\[
Y_{t,i}^{(j)} \sim \text{Bernoulli}(\pi_{t,i}^{(j)}),
\]

\[
\pi_{t,i}^{(j)} = \frac{e^{\xi_{t,i}^{(j)}}}{1 + e^{\xi_{t,i}^{(j)}}},
\]

\[
\xi_{t,i}^{(j)} = \xi_{t-1,i}^{(j)} + \tilde{\theta}_{t,i}^{(j)} + w_{t,i}^{\xi^{(j)}}, \quad w_{t,i}^{\xi^{(j)}} \sim N(0, W_{\xi}^{(j)})
\]

\[
\tilde{\theta}_{t,i}^{(j)} = \lambda_{t,i}^{(j)} \tilde{\theta}_{t-1,i}^{(j)} + \beta X_{t,i}^{(j)} + w_{t,i}^{\theta^{(j)}}, \quad w_{t,i}^{\theta^{(j)}} \sim N(0, W_{\theta}^{(j)})
\]

Here \( \lambda_{t,i}^{(j)} \) represents the decay factor that determines the contribution of previous state values to the current one for each subsystem. In this factor, we take into account the different degradation rates that each subsystem is subject to experience. Therefore, low degradation system may rely more on previous state values whereas high degradation systems rely only on the previous and current state.

The vector \( X_{t,i}^{(j)} \) is constructed from the patient’s observed lab test, vital sig-
nals, text, and demographics (features) for each particular subsystem.

In the later subsections of this chapter, we explain how to extract the features of each subsystem. Also, we indicate the method to model and imputed the missing values for each subsystem feature vector. The value of $\beta^{(j)}$ represents the vector of regression coefficients we use to combine the observed features. Note that for our proposed method, the dimension of $\beta^{(j)}$ is different for each subsystem.

$W^{(j)}_{\xi}$ and $W^{(j)}_{\theta}$ represent the evolution variances of $\xi^{(j)}$ and $\tilde{\theta}^{(j)}$ respectively.

The value of $\xi^{(j)}_{t}$ reflects the log-odds effect on the propensity of subsystem failure $\pi^{(j)}_{t,i}$ by previous observed features contained in the state $\theta^{(j)}_{t,i}$. To illustrate this effect, we calculate the impact of the user’s features $X^{(j)}_{t,i}$ observed at time $t$ for the subsystem $j$ and then aggregate them into the state $\theta^{(j)}_{t,i}$ after $k$ steps assuming no other value of $[X^{(j)}_{t+1,i} \ldots X^{(j)}_{t+k,i}]$ is observed. This impact is determined by the following forecast function:

$$\xi^{(j)}_{t+k} = \sum_{r=0}^{k} \lambda^{(j)} r \tilde{\theta}^{(j)}_{t} = \tilde{\theta}^{(j)}_{t} (1 - \lambda^{(j)} k + 1) / (1 - \lambda^{(j)})$$

(7.5)

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 the propensity of subsystem failure even when no measurements are available at a given time $t + k$. In addition, this effect does not decrease over time, as opposed to the state evolution based on observed features $\tilde{\theta}^{(j)}_{t}$. Each time there are new observations available, the value of the effect $\xi^{(j)}_{t,i}$ is updated using equation 3.
We can increase or decrease the propensity of subsystem failure using the observed features $X_{t,i}^{(j)}$ by allowing $\tilde{\theta}_{t,i}^{(j)}$ to have both positive and negative values.

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

$$Y_{t,i}^{(j)} \sim \text{Bernoulli}(\mu_{t,i}^{(j)}),$$

$$\mu_{t,i}^{(j)} = \frac{e^{F'\theta_{t,i}^{(j)}}}{1 + e^{F'\theta_{t,i}^{(j)}}},$$

$$\xi_{t,i}^{(j)} = F'\theta_{t,i}^{(j)},$$

$$\theta_{t,i}^{(j)} = G^{(j)}\tilde{\theta}_{t-1,i}^{(j)} + [0, \beta^{(j)}X_{t,i}^{(j)}]' + w_t$$

$$F' = [1, 1]$$

$$G^{(j)} = \begin{bmatrix} 1 & 0 \\ 0 & \lambda^{(j)} \end{bmatrix}$$

$$w_t \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} W_{\xi}^{(j)} & 0 \\ 0 & W_{\theta}^{(j)} \end{bmatrix}\right)$$

(7.6)

Figure 7.1 shows the graphical model of this framework. The colored circles represent the variables that are observed in the model. The non-colored circles represent the latent variables and model parameters that need to be inferred. The learning across multiple users is reflected through the estimated parameters $\Phi^{(j)}$, which are defined as:

$$\Phi^{(j)} = \lambda^{(j)}, W_{\theta}^{(j)}, W_{\xi}^{(j)}, \beta^{(j)}.$$  

(7.7)

This representation is flexible enough to expand the model and to incorporate different weighting vectors $\beta^{(j)}$ for different patient groups with a particular age range.

We model each subsystem independently as illustrated in figure 7.1. Then we combine all the states into an aggregated one as it is shown in figure 7.2. Thus, we estimate the model parameters $\Phi^{(j)}$ for each system independently.

The probability likelihood for observation $Y_{t,i}^{(j)}$, described in Equation 1-4, fol-
Figure 7.1: Graphical model of the subsystem failure state

Figure 7.2: Graphical model of the estimation of the probability of mortality using the subsystem propensity failure

follows a Bernoulli distribution. Thus we transform the subsystem failure propensity from $\pi_{t,i}$ 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 subsystem state, which evolves over time, in contrast to static classification models and sequential state models. This approach allows us to predict future values of the state using the current and past observations. Consequently, we
are able to estimate the propensity of the subsystem to failure and predict its evolution using the forecast function of the latent state of the subsystem $j$.

Algorithm 8 describes the fitting steps for each subsystem of the proposed model. The evolution of the state $\theta_{t,i}^{(j)}$, and the parameter estimation steps are discussed in the following subsections.

### 7.1.1 Filtering Update and Model Inference

In this subsection, we develop the filtering update method to incorporate binary observations for each subsystem $j Y_{t,i}^{(j)}$. We sequentially model the propensity of the subsystem to fail and its evolution over time. Then, we outline the backward smoothing recursion required to estimate the optimal model parameters $\Phi^{(j)}$ which take into account future uncertainty. Finally, we describe the method we use to estimate the model parameters which is based on Expectation Maximization.

#### 7.1.1.1 Kalman Filtering Update

We use the Kalman filtering equations to update the latent state using Dynamic Linear Models (DLMs) when series observations become available, $p(\theta_{t,i}^{(j)}|\Phi^{(j)}, D_{1:t,i}^{(j)})$ for $t = 1, \ldots, T$ where $D_{1:t,i}^{(j)} = [X_{1:t,i}^{(j)}, Y_{1:t,i}]$ is the observed data of the subsystem $j$ up to time $t$ and the model parameters $\Phi^{(j)}$. Here, the observations are assumed to be continuous and normally distributed as defined in the standard DLM [92, 73]. However, to model the propensity of subsystem failure, we transform $\pi_{t,i}^{(j)}$ to $\xi_{t,i}$ using the logistic transformation. Therefore, $\xi_{t,i}^{(j)}$ follows a logistic normal distribution. The main chal-
lenge in using this distribution is the lack of a conjugate prior distribution for $\xi_{t,i}^{(j)}$, and consequently for $\theta_{t,i}^{(j)}$, which prevents us from estimating this probability in closed form.

To overcome this challenge, we approximate the filtering update distribution by a normal distribution using the Laplace approximation [29]. Here, $\theta_{t,i}^{(j)}$ is approximated to a Normal using the Maximum A Posteriori (MAP) distribution $\theta_{t,i}|y_{t,i}$. The we consider the MAP estimate as the mean and the Hessian of the log posterior distribution evaluated at the MAP as the variance.

$$\theta_{t,i}|y_{t,i} \approx p(\theta_{t,i})p(y_{t,i}|\theta_{t,i})$$

(7.8)

We use the evolution of the state as prior distribution for $p(\theta) \sim N(a_{t,i}^{(j)}, R_{t,i}^{(j)})$. $p(y_{t,i})$ follows a bernoulli distribution. Using the Laplace approximation this leads to:

$$\ln\theta_{t,i}|y_{t,i} \approx -\ln(1 + \exp\{-Y_{t,i}^{(j)}F\theta_{t,i}^{(j)}\} - \frac{1}{2}(\theta_{t,i}^{(j)} - a_{t,i}^{(j)})R_{t,i}^{(j)-1}(\theta_{t,i}^{(j)} - a_{t,i}^{(j)})$$

(7.9)

Let $a_{t,i}$ be the predictive mean and $R_{t,i}^{(j)}$ be the predictive variance given $X_{t,i}^{(j)}, D_{t-1,i}^{(j)}$ for the subsystem $j$ of a patient $i$ at the time $t$ where:

$$a_{t,i}^{(j)} = G^{(j)}m_{t-1,i} + \beta^{(j)}X_{t,i}^{(j)}$$

$$R_{t,i}^{(j)} = G^{(j)}C_{t,i}^{(j)}G' + W$$

(7.10)

where $m_{t-1,i}$ and $C_{t,i}$ is mean and the variance of the state obtained from the Kalman Filter.

We use Newton-Raphson iterative method to find the MAP since the closed form maximization is not feasible for this case. By letting $\theta_{t,i}^{MAP(j)}$ and $H(\theta_{t,i}^{MAP(j)})$ be
the MAP state estimate and the Hessian respectively, filtering update process leads to
following state expressions:

\[
\begin{align*}
\theta^{(j)}_{t,i} | D^{(j)}_{1:t,i}, \Phi^{(j)} & \sim N(m^{(j)}_{t,i}, C^{(j)}_{t,i}), \\
m^{(j)}_{t,i} & = \theta^{(j)\text{MAP}}_{t,i}, \quad C^{(j)}_{t,i} = -H^{-1}(\theta^{(j)\text{MAP}}_{t,i})
\end{align*}
\] (7.11)

where \(\theta^{(j)}_{t,i} | D^{(j)}_{1:t,i}\) is the state value for the patient \(i\) at time \(t\) given the complete observed
data \(D^{(j)}_{1:t,i}\) for the subsystem \(j\) and the model parameters \(\phi^{(j)}\).

\[
\begin{align*}
\theta^{(j)\text{MAP}}_{t,i} & = \arg \max_{\theta^{(j)}_{t,i}} -\ln(1 + \exp\{-Y^{(j)}_{t,i} F^{T} \theta^{(j)}_{t,i}\} - \frac{1}{2}(\theta^{(j)}_{t,i} - a^{(j)}_{t,i})^{T}R^{(j)}_{t,i}^{-1}(\theta^{(j)}_{t,i} - a^{(j)}_{t,i}) \\
\frac{\partial l(\theta^{(j)}_{t,i})}{\partial \theta^{(j)}_{t,i}} & = \frac{FY^{(j)}_{t,i}}{1 + \exp\{Y^{(j)}_{t,i} F^{T} \theta^{(j)}_{t,i}\}} - R^{(j)}_{t,i}^{-1}(\theta^{(j)}_{t,i} - a^{(j)}_{t,i}), \\
\frac{\partial^{2} l(\theta^{(j)}_{t,i})}{\partial \theta^{(j)2}_{t,i}} & = \frac{-FF'}{(1 + \exp\{-Y^{(j)}_{t,i} F^{T} \theta^{(j)}_{t,i}\})(1 + \exp\{Y^{(j)}_{t,i} F^{T} \theta^{(j)}_{t,i}\})} - R^{(j)}_{t,i}^{-1}
\end{align*}
\] (7.12)

Based on these derivatives, the MAP update for the Newton-Raphson method
is defined as follows:

\[
\begin{align*}
\theta^{(j)}_{t,i} | D^{(j)}_{1:t,i} & \sim N(m^{(j)}_{t,i,j}, C^{(j)}_{t,i,j}), \\
m^{(j)}_{t,i} & = \theta^{(j)\text{MAP}}_{t,i}, \quad C^{(j)}_{t,i} = -H^{-1}(\theta^{(j)\text{MAP}}_{t,i})
\end{align*}
\] (7.13)

In this manner, we learn and refresh the state distribution from time \(t - 1\) to \(t\) using the
discrete output observations. This estimation replaces the standard Kalman filtering
equations to model the propensity of subsystem failure as a latent state.
### 7.1.1.2 Forward Filtering Backward Smoothing

We use Forward Filtering Forward Filtering Backward Smoothing (FFBS) method to account for future uncertainty in the estimation of the state. With this method, we estimate the expected value of the states $\theta_{t,i}^{(j)}$ given the parameters in a DLM $\Phi^{(j)}$ [73, 31].

First, we estimate mean $m_{t,i}^{(j)}$ and $C_{t,i}^{(j)}$ variance of the hidden states distribution $p(\theta|D_{1:t,i}^{(j)}) \sim N(m_{t,i}^{(j)}, C_{t,i}^{(j)})$ given the observations up to time $t$ using the filtering equations described in previous subsection. Once we have estimated these values for the entire patient path, we obtain the smoothing mean $s_{T_i,i}^{(j)}$ and variance $S_{T_i,i}^{(j)}$ of the state variables at time $T_i$, $\theta_{T_i,i}^{(j)}|\theta_{D_{1:T_i,i}}^{(j)} \sim N(s_{T_i,i}^{(j)}, S_{T_i,i}^{(j)})$. Conditional on this value, we estimate the smoothing mean $s_{T_{i-1},i}^{(j)}$ and variance $S_{T_{i-1},i}^{(j)}$ the state variables at time $T_i - 1$ (backwards).

By combining these two steps, we guarantee the construction of an entirely dynamic model with feedback. Autoregressive and some of the Hidden Markov-based models are trained using open loop feedback (without any feedback about the feature). We can fit our proposed model in this framework by only using the mean and variance estimated in the Forward Filtering step (FF) described in subsection 7.1.1.1.

To achieve numerical stability in the estimation, we use estimate the values of $s_{t,i}^{(j)}$ and $S_{t,i}^{(j)}$ by using the approach detailed in [101]. This method is based in Singular Value Decomposition (SVD). Algorithm 9 depicts the steps needed to perform FFBS in our proposed model.
7.1.1.3 Subsystem Model Parameter Estimation

We estimate the value of the model parameters using an Expectation Maximization (EM) based method for each subsystem we analyze. The parameter values are determined for each subsystem independently. At this stage, we assume there is no interdependency between subsystems.

In the first step (E-step), we estimate the expected state value for every subsystem for each time step of the patient path. We perform this task using the Forward Filtering Backward Sampling (FFBS) algorithm, which we explained in the previous subsection. In the M-step we estimate the value of the parameters that maximizes the likelihood defined as follows:

\[
p(\theta^{(j)}, \Phi^{(j)}|D^{(j)}_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{P} p(\theta^{(j)}_{t,i}, \theta^{(j)}_{t-1,i})p(\theta^{(j)}_{t,i} | \Phi^{(j)}, D^{(j)}_{T_i}) \tag{7.15}
\]

The log-likelihood function is concave given the set of parameters \( \Phi^{(j)} \). As a consequence, the maximum likelihood estimate (MLE) is unique.

To estimate the value of each parameter, 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 [31] and after some algebra, we obtain the M-step update expressions given the current value of parameters as follows:
\[ G^{(j)}_{\text{new}} = \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t-1,i}^{(j)}, \theta_{t,i}^{(j)}] \right]'^{-1} \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t-1,i}^{(j)}, \theta_{t,i}^{(j) \prime}] \right] \]

\[ E[\theta_{t-1,i}^{(j)}, \theta_{t,i}^{(j) \prime}] = m_{t-1,i}^{(j)} s_{t,i}^{(j)} + L_{t-1,i} \left( S_{t,i}^{(j)} + (s_{t,i}^{(j)} - a_{t,i}^{(j)}) s_{t,i}^{(j) \prime} \right) - \beta^{(j)} X_{t,i}^{(j)} s_{t,i}^{(j) \prime} \]

\[ E[\theta_{t-1,i}^{(j)}, \theta_{t,i}^{(j) \prime}] = S_{t,i}^{(j)} s_{t,i}^{(j)} \]

\[ W^{\text{new}} = \frac{1}{\sum_{i=1}^{P} T_i} \sum_{i=1}^{P} \sum_{t=1}^{T_i} E[\theta_{t,i}^{(j)}, \theta_{t,i}^{(j) \prime}] - E[(G^{(j)} \theta_{t-1,i}^{(j)} + \beta^{(j)} X_{t,i}^{(j)}) \theta_{t,i}^{(j) \prime}] \]

\[ E[\theta_{t,i}^{(j)}, \theta_{t,i}^{(j) \prime}] = S_{t,i}^{(j)} s_{t,i}^{(j)} \]

\[ E[G^{(j)} \theta_{t-1,i}^{(j)}, \theta_{t,i}^{(j) \prime}] = G \left[ m_{t-1,i}^{(j)} s_{t,i}^{(j)} + L_{t-1,i} \left( S_{t,i}^{(j)} + (s_{t,i}^{(j)} - a_{t,i}^{(j)}) s_{t,i}^{(j) \prime} \right) \right] \]

\[ E[\beta^{(j)} X_{t,i}^{(j)} \theta_{t,i}^{(j) \prime}] = \beta^{(j)} X_{t,i}^{(j)} s_{t,i}^{(j)} \]

\[ \beta^{\text{new}} = \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} (s_{t,i}^{(j)} - G^{(j)} s_{t-1,i}^{(j) \prime}) X_{t,i}^{(j)} \right] \left[ \sum_{i=1}^{P} \sum_{t=1}^{T_i} X_{t,i}^{(j) \prime} X_{t,i}^{(j)} \right]^{-1} \]

where:

\[ L_{t-1,i}^{(j)} = C_{t-1,i}^{(j)} G^{(j) \prime} R_{t,i}^{(j)} \]

\[ p(\theta_{t,i} | D_{t,i}^{(j)}) \sim N(a_{t,i}^{(j)}, R_{t,i}^{(j)}) \]

\[ R_{t+1,i}^{(j)} = G^{(j)} C_{t,i}^{(j)} G^{(j) \prime} + W_{t,i}^{(j)} \]

\[ a_{t+1,i}^{(j)} = G m_{t,i}^{(j)} + \beta^{(j)} X_{t+1,i}^{(j)} \]

\[ p(\theta_{t,i} | D_{t,i}^{(j)}) \sim N(a_{t,i}^{(j)}, C_{t,i}^{(j)}) \]

\[ p(\theta_{t,i} | D_{t,i}^{(j)}) \sim N(s_{t,i}^{(j)}, S_{t,i}^{(j)}) \]

The values of \( m_{t,i}^{(j)} \) and \( C_{t,i}^{(j)} \) for \( t = 1 \ldots T_i \) are calculated during the time series filtering step described in subsection 7.1.1.1. Given these estimations, backward smoothing is performed, based on standard Kalman smoothing recursion described in
subsection 7.1.1.2, to obtain $s^{(j)}_{t,i}$, $S^{(j)}_{i,t}$ for each subsystem. This process represents the E-step of the EM algorithm. The values of $a^{(j)}_{t,i}$ and $R^{(j)}_{t,i}$ represent the predictive expected state and the predictive state variance of the subsystem $j$.

7.2 Estimation of the Probability of Mortality

We estimate the probability of patient mortality $P(M_{t,i})$ at every time step $t$ as a combination of the failure propensity scores for all the subsystems $j = 1 \ldots J$ using logistic regression. Therefore the probability of mortality for the patient $i$ at each time step $t$ is defined as follows:

$$P(v_{t,i} = 1 | \mu^{(1)}_{t,i} \ldots \mu^{(J)}_{t,i}) = \frac{e^{\sum_{j=1}^{J} \mu^{(j)}_{t,i}}}{1 + e^{\sum_{j=1}^{J} \mu^{(j)}_{t,i}}} \quad (7.20)$$

where $\mu^{(j)}_{t,i}$ is the failure propensity score for subsystem $j$. These scores are previously estimated by the model described in subsection 5.3. In this framework, we detect which combination of subsystem failure leads to the patient dead. In addition, we improve dynamic the estimation of the patient probability of mortality using these physiological subsystems, which are provided by medical experts hierarchically.

7.3 Feature Extraction and Missing Values Estimation

In this section, we explain the process we follow to extract the features used by each subsystem to estimate its propensity to fail. We obtain three types of features: numeric, text-based and topic based. Algorithm 7 shows the feature extraction process.
7.3.1 Numerical Features Extraction

We extract features for 5 physiological subsystems: vital signs, cardiovascular, respiratory, renal and hepatic. For each subsystem, we use a set of numerical features from vital signal readings, labs and procedures provided by nurses and physicians. For instance, for the cardiovascular subsystem we select blood pressure, heart rate, magnesium, cholesterol, cardiac enzymes levels among other features.

When we analyze the extracted features used in our model, we note that 70% of them are used by the Apache Score III [46] and by the SAPS II score [44].

We identify that some of the selected features are considered to have a bimodal distribution such as blood pressure. 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 [46].

We update the value of the features each time there is a new reading. In later subsections of the chapter, we explain how we treat missing values in our model using the subsystem information and the non-ignorable nature of the data.

7.3.2 Text Features Extraction

Standard approaches to estimating the probability of subsystem failure and patient mortality such as Apache III and SAPS II scores, do not incorporate text information. The text entries found in an EMR mainly consist of nurse’s entries, procedures reports, admission and discharge information, among others. The primary challenge to incorporate text data in prediction models is the complexity and the sparsity of the
vocabulary features.

The text data, which we integrate into our model, consists of a sequence of entries, each with an assigned timestamp. Thus, we can construct a time series of text features. Figure 5.2 depicts the text-based feature extraction process.

7.3.3 Missing Features Value Estimation

The proposed framework mentioned above assumes that we observe all the patient’s features at each point in time. When the value is not seen, we indicate they are missing. 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.

Also, the lack of some of the observed features implies that the patient does not require particular labs or procedures to be performed. For instance if the patient does not have a heart attack, then the cardiac enzymes lab should not be performed. Thus, one portion of the patient features are non-ignorable in nature (Not missing at random) especially the some of the labs performed.

We integrate this knowledge into the model by dividing the features into two parts. In the first part, we include features that have ignorable data (such as vital signs readings). In the second part, we include features that have non-ignorable data (some of the labs performed for a particular subsystem).

When the features values are not observed in the ignorable data part, we carry the previously observed value. If there is not previous value observed, we impute its
value using multiple imputation with the other features as covariates. On the other hand, for the missing values in the non-ignorable features, we assume no contribution for this particular feature into the model. Thus, we indicate that the feature is missing and calculate the value state without the feature value. In this manner, we reduce the noise introduced by the missing values in the estimation process.

7.4 Results

In this section, we depict the experimental settings to test and validate the proposed model. In addition, we report our experimental results of our method with respect to other dynamic methods used in the literature. Finally, we discuss the impact of our proposed model in a real scenario.

We test the proposed framework by estimating the probability of mortality for a patient who is inside the Intensive Care Unit (ICU) in a hierarchical form. First, we calculate the probability of subsystem failure for 5 systems commonly checked by physicians and nurses inside the ICU. These are vital signs, cardiovascular, respiratory, renal and hepatic. Then, we combine the estimated probability of failure for each patient at each time step. Then, we determine the probability of mortality of a patient (in-hospital mortality) and trigger and alarm if this probability is larger than a certain threshold. The accurate and opportune prediction of this probability is critically important to assign medical resources efficiently and to aid doctors and nurses ahead of time.
We predict each of the patient’s subsystem failure probability using the information available in his/her EMR. We obtain the EMRs from the MIMIC II data set [75]. This dataset contains text and numerical information that describes procedures, medications and vital signs readings for a given patient during his/her stay in the ICU. MIMIC II is composed of medical records from over 30,000 patients. We validate our method using 15,000 patients who are selected randomly. To compare our proposed model, we select only the adult patients (18 years old or older) from the selected subset without excluding any patient disease; this data consists of 11,648 health records.

7.4.1 Experimental Settings

We construct a time series sequence of each selected feature events using the registered time stamp in three-hour increments for each of the subsystems we model. Therefore, we have 5 sets of different features. 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 about the patient admission to the ICU \( t = 0 \). Thus, we construct the relative series of events using the anonymized data and perform the analysis.
Algorithm 8 Proposed Method

Define $J$ to be the number of physiological subsystems to model

for $j \leftarrow 1$ to $J$ do

Extract text and topic Features of subsystem $j$

Construct time series

Impute Missing Values

Use an initial guess of the parameters $\Phi^{(j)} = (\lambda^{(j)}, W^{(j)}_\xi, W^{(j)}_\theta, \beta^{(j)})$

Define $P$ to be the number of patients in the dataset

repeat

for $i \leftarrow 1$ to $P$ do

Define $T_i$ to be the number of time steps of the series for a patient $i$

for $t \leftarrow 1$ to $T_i$ do

Estimate the value of $m^{(j)}_{t,i}$ and $C^{(j)}_{t,i}$ using Filtering Equations described in subsection 7.1.1.1

end for

for $t \leftarrow T_i$ to 1 do

Estimate the expected value $s^{(j)}_{t,i} = E(\theta^{(j)}_{t,i}|D^{(j)}_{1:T_i})$ and variance $S^{(j)}_{t,i} = var(\theta^{(j)}_{t,i}|D^{(j)}_{1:T_i})$ of the hidden states using FFBS algorithm described in subsection 5.4.2

end for

end for

Estimate the parameters $\Phi^{j}$ that maximize the likelihood $P(\theta, \Phi|D^{(j)}_{1:T_i,1:P})$

until Convergence

end for

Estimate weights $\alpha_j$ for each subsystem $j = 1 \ldots J$ in the aggregated model to estimate the probability of mortality using the propensity values $\mu^{(j)}_{t,i}$ for $t = 1 \ldots T_i$, $i = 1 \ldots P$ and $j = 1 \ldots J$
Algorithm 9 Forward Filtering Backward Smoothing

Estimate \( p(\theta_{t,i}^{(j)})|\Phi, D_{1:T,i}^{(j)}) \sim N(m_{t,i}^{(j)}, C_{t,i}^{(j)}) \) for \( t = 1, \ldots, T \) as discussed in subsection 7.1.1.2.

Estimate \( \theta_{T,i,i}^{(j)}|D_{1:T,i}^{(j)} \sim N(m_{T,i,i}^{(j)}, C_{T,i,i}^{(j)}) \)

for \( t \leftarrow T - 1 \) to 1 do

Estimate \( \theta_{t,i,i}^{(j)}|\theta_{t+1,i,i}, D_{1:T,i}^{(j)} \sim N(s_{t,i,i}, s_{t,i,i}) \) (Backward Smoothing)

\[
s_{t,i,i} = m_{t,i} + C_{t,i}G'R_{t+1,i}(s_{t+1,i} - a_{t+1,i})
\]

\[
S_{t,i,i} = C_{t,i} - C_{t,i}G'R_{t+1,i}(R_{t+1,i} + S_{t+1,i})R_{t+1,i}^{-1}GC_{t,i}
\]

\[
R_{t+1,i} = GGC_{t,i}G' + W_{t,i}
\]

\[
a_{t+1,i} = Gm_{t,i} + \beta X_{t+1,i}
\]

end for
Table 7.1: Performance of the 3 variants of our model for each subsystem A. Vital Signs B. Cardiovascular C. Respiratory. D. Renal E. Hepatic

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>24 hours</th>
<th>48 hours</th>
<th>72 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Sensitivity</td>
<td>Specificity</td>
</tr>
<tr>
<td>Numeric Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem A</td>
<td>0.8551</td>
<td>0.8404</td>
<td>0.8796</td>
</tr>
<tr>
<td>Subsystem B</td>
<td>0.8592</td>
<td>0.8356</td>
<td>0.7688</td>
</tr>
<tr>
<td>Subsystem C</td>
<td>0.8733</td>
<td>0.8466</td>
<td>0.8747</td>
</tr>
<tr>
<td>Subsystem D</td>
<td>0.8640</td>
<td>0.8423</td>
<td>0.8030</td>
</tr>
<tr>
<td>Subsystem E</td>
<td>0.8604</td>
<td>0.8505</td>
<td>0.8863</td>
</tr>
<tr>
<td>Numeric + Text Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem A</td>
<td>0.9203</td>
<td>0.9146</td>
<td>0.9012</td>
</tr>
<tr>
<td>Subsystem B</td>
<td>0.8391</td>
<td>0.8189</td>
<td>0.7589</td>
</tr>
<tr>
<td>Subsystem C</td>
<td>0.9454</td>
<td>0.9346</td>
<td>0.7931</td>
</tr>
<tr>
<td>Subsystem D</td>
<td>0.9218</td>
<td>0.9234</td>
<td>0.5886</td>
</tr>
<tr>
<td>Subsystem E</td>
<td>0.7763</td>
<td>0.8391</td>
<td>0.7016</td>
</tr>
<tr>
<td>Numeric + Text + Topic Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem A</td>
<td>0.9590</td>
<td>0.9550</td>
<td>0.9780</td>
</tr>
<tr>
<td>Subsystem B</td>
<td>0.9036</td>
<td>0.9054</td>
<td>0.9719</td>
</tr>
<tr>
<td>Subsystem C</td>
<td>0.9471</td>
<td>0.9323</td>
<td>0.9155</td>
</tr>
<tr>
<td>Subsystem D</td>
<td>0.9262</td>
<td>0.9142</td>
<td>0.9194</td>
</tr>
<tr>
<td>Subsystem E</td>
<td>0.9440</td>
<td>0.9432</td>
<td>0.8138</td>
</tr>
</tbody>
</table>
We obtain the labels of subsystem failure by using the ICD9 codes attached to every patient. These codes describe if there is a subsystem failure during the entire patient stay in the ICU. We observe that 90% of the patients have 2 or less failing subsystems. However, people who die inside the ICU have 2 or more failing subsystems (86%). Therefore, the correct prediction of subsystem failure could improve the performance in the estimation of mortality.

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. The unobserved features are related to the systems that are not failing. This corroborates our hypothesis that the lack of observed features in a particular subsystem can help us to determine which subsystems are prone to fail and which ones are not.

To extract the text features, we consider each text entry to be a document. There is 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 [16] model using all the text entries and $K = [50, 75, 100]$ topics. The resulting topics provide a better context that aids to improves the performance of the classification task. We determine that the best quality of topics is achieved on 75 topics using empirical likelihood as comparative measure [16]. After estimating the topic mixture for each document, we remove the background topics as described above. We retain 65 topics after this step. Then we determine the classification output using topics and term features for each subsystem $j$ for every text entry.
We validate our model using a five-cross validation. Here, we select randomly 80% percent of the patients to be the training set, and we use the remaining 20% as the test set. We fit each subsystem prediction model using three different set of features: numerical; numerical + term based; numerical + term based + topic modeling based.

To estimate the probability of mortality, we combine the estimation of the probability of failure from each subsystem model. We observe that approximately 34% of the selected patients stayed 24 hours or less. This fact means that standard practices to predict the probability of mortality such as the Apache Score [46] and SAPS II [44] cannot be calculated for those patients.

Then, we compare these results with quasi-dynamic models 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 employed in the literature: Apache III Score [46], SAPS II [44] and the results provided by Ghassemi et al. [32]. Finally, we compare the proposed model by using all the features into a single aggregated state (where no subsystem information is included). Our goal is to test what is the performance increase of using subsystem information compared to all the features compiled into a single aggregated state. Our subsystem model has similar performance compared to the aggregated model in terms of AUC. However, our subsystem model allows us to further segment patients according to a particular subsystem failure, making the results more interpretative.
Table 7.2: Performance of the 3 variations of our subsystem model, 3 variations of the aggregated state, 3 different methods used in the literature and 2 static classification methods with sliding window

<table>
<thead>
<tr>
<th></th>
<th>24 hours</th>
<th></th>
<th></th>
<th>48 hours</th>
<th></th>
<th></th>
<th>72 hours</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sensitivity</td>
<td>Specificity</td>
<td>AUC</td>
<td>Sensitivity</td>
<td>Specificity</td>
<td>AUC</td>
<td>Sensitivity</td>
<td>Specificity</td>
</tr>
<tr>
<td>Apache III</td>
<td>0.6925</td>
<td>0.1090</td>
<td>0.6769</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SAPS II</td>
<td>0.6890</td>
<td>0.1393</td>
<td>0.6239</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Random Forests</td>
<td>0.6208</td>
<td>0.1084</td>
<td>0.7421</td>
<td>0.6926</td>
<td>0.1004</td>
<td>0.7460</td>
<td>0.692</td>
<td>0.0910</td>
</tr>
<tr>
<td>Ghassemi2014</td>
<td>0.638</td>
<td>0.85</td>
<td>0.8400</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.6559</td>
<td>0.1622</td>
<td>0.6367</td>
<td>0.5067</td>
<td>0.3243</td>
<td>0.6215</td>
<td>0.637</td>
<td>0.1916</td>
</tr>
<tr>
<td>Agg. State Numerical Features</td>
<td>0.7038</td>
<td>0.5024</td>
<td>0.7606</td>
<td>0.7410</td>
<td>0.5564</td>
<td>0.6400</td>
<td>0.7310</td>
<td>0.62567</td>
</tr>
<tr>
<td>Agg. State Text and Numerical Features</td>
<td>0.7648</td>
<td>0.6924</td>
<td>0.7806</td>
<td>0.8071</td>
<td>0.4764</td>
<td>0.7445</td>
<td>0.7010</td>
<td>0.6567</td>
</tr>
<tr>
<td>Agg. State Topic, Text and Numerical Features</td>
<td>0.7885</td>
<td>0.7905</td>
<td>0.8657</td>
<td>0.7822</td>
<td>0.7685</td>
<td>0.7985</td>
<td>0.7468</td>
<td>0.7992</td>
</tr>
<tr>
<td>Subsystem Numerical Features</td>
<td>0.865</td>
<td>0.4688</td>
<td>0.7644</td>
<td>0.85002</td>
<td>0.5538</td>
<td>0.6662</td>
<td>0.8650</td>
<td>0.59567</td>
</tr>
<tr>
<td>Subsystem Text and Numerical Features</td>
<td>0.8933</td>
<td>0.626</td>
<td>0.7972</td>
<td>0.87266</td>
<td>0.4883</td>
<td>0.7445</td>
<td>0.8730</td>
<td>0.498</td>
</tr>
<tr>
<td>Subsystem Topic, Text and Numerical Features</td>
<td>0.8685</td>
<td>0.7422</td>
<td>0.8660</td>
<td>0.785</td>
<td>0.785</td>
<td>0.8085</td>
<td>0.7968</td>
<td>0.7392</td>
</tr>
</tbody>
</table>
7.4.2 Experimental Results

We fit the parameters of our proposed model for each subsystem using the Expectation Maximization approach described in section 7.1.1 using 50 iterations for the 3 sets of features. We observe that the average estimated value of $\lambda$ for all the subsystems is around 0.01 for numerical features and text and topic features for all the systems. This value of $\lambda$ imply that the current effect of the observations taken at time $t$ is reduced to 10% after 4 steps respectively ($t + 3$ and $t + 4$). These step numbers are equal to the common delays between patient assessments inside the ICU (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 for each subsystem. When we use topic models to classify subsystem failure, we notice that in the topic-based features at least one text entry indicates that the system is failing (around 70% of all the text entries were correctly classified). This signal is not evident when using term based features only. In this context, the classification based on terms for failing systems may not have any indication of failure. This fact corroborates our hypothesis that topic based features result in better classification features than term based features.

Table 7.1 shows the performance in detecting subsystems failure for $t = 24, 48$ and 72 hours. Here we observe that the use of a dynamic setting provide a good prediction rate for all the subsystems modeled. We also observe that the use of topic and text features improve the prediction of system failure significantly in all the subsystems.
except the renal where the improvement is almost null. We observe that the performance of the model decreases after 24 hours. This result is consistent with findings of authors in [25]. They indicate that the initial predictions are better estimates than the subsequent ones. The best form to assess system failure in the subsequent readings is with the gradient of probability compared to the absolute value.

Table 7.2 shows the performance of our probability of mortality estimation using the combination of subsystem failure propensity scores. We also show a comparison between our proposed framework with the three variants we test. In addition, we show the performance of static classification methods such as Random Forest with 50 trees and Naive Bayes with a sliding window, for 24, 48 and 72 hours. Finally, we compared our method with the Apache III and the SAPS II scores and the results reported by Ghassemi [32] which are calculated 24 hours after the patient is admitted to the ICU.

Note that our method provides estimates every time there is a new patient observation compared to the other methods that are calculated using a 24-hour window. This allows us to perform dynamic interventions (medications and treatments) to modify the current state estimate. To make our method comparable, we report results at 24, 48 and 72 hours.

Table 7.2 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. As we observe, our method that combines numerical, term based and topic based features has the highest AUC for 24 and 48 hours with respect to other methods. Our proposed model’s AUC is at least 3.05% higher with respect to Ghassemi et al. (0.8660 vs. 0.8400) in 24
hours, and by 7.04% with respect to Random Forest. We note that our model clearly outperforms all the other methods tested in specificity.

Our method shows better performance even when using only numerical features. This is due to the aggregation dynamic subsystem failure in previous times on the current estimation of the probability of mortality. Our method, with only numerical features, obtains better performance in terms of AUC than Apache Score (0.6925) and SAPS II (0.6899). Also, our method can 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 curve. We observe that all three variants of our method have better performance consistently 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. We hypothesize 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. This decrease is due to the variability among patients effect previously explained in the subsystem failure. (from 0.8657 at 24 hours to 0.8085 at 48 for the numerical+term based+topics
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 data leads to very poor results that are not comparable with the other methods shown in Table 7.2. Thus, we do not report the results of this technique.

We do not compare our method with HMM-based models. Some of these models predict disease progression by addressing specific comorbidities as latent discrete states and do not discuss the probability of mortality [88]. Other HMM-based approaches are used to complete missing features or to analyze the risk of specific diseases [15]. We do not include any of these methods in our table because these methods address other problems and are less powerful.

7.5 Discussion

In this chapter, we have proposed a framework to exploit the dynamical and heterogeneous information from the Electronic Medical Records of patients who are admitted to the ICU to predict a subsystem failure. Our method provides a fully dynamic framework that takes into account future uncertainty by training the model using the complete patient path from admission to discharge/death with high performance.

We note that some of the mortality prediction methods already used in the literature are fitted by taking into account the existence of body subsystems. They ac-
count for this in a limited form by using features regarding all the subsystems. However, they aggregate all the features in a single probability value. This aggregation does not anticipate the fact the some subsystems can be degraded faster than others. However, the degradation in each subsystem is included in a limited form by using the worst case value to predict the patient mortality. Our method explicitly predicts the probability of failure for each patient subsystem using a different rate for each subsystem.

We expand the aggregated model to provide a more interpretable framework. Our goal is to aid the doctors and medical staff to take decisions about the evolution of a patient, taking into account his personal evolution and other patients’ health path.

We treat the patient 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. We found that the absence of value in some of the labs, actually carries valuable information about the state of the patient.

We notice that text information results in reliable features when the text entries are concrete as opposed to a combination of different doctor notes such as the admission notes. Therefore to incorporate effectively this type of entries we need to segment further the document to separate the different kind of information each segment would provide.

Further work includes taking advantage of the dynamic nature of the model. We can predict not only the presence of subsystem failure but also to predict how likely the patient would be able to recover (no recovery, slow recovery with sequels or fast recovery with no sequels).

In the currently proposed framework, we estimate the probability of mortality
as a linear combination of subsystem failure propensity scores using logistic regression without any interactions among systems. Future work would include exploring different classification methods and feature combination that account for the interaction of the various subsystems to better predict the probability of mortality.
Chapter 8

Conclusion

The research compiled in this thesis is based on the use of Statistical Topic Models in Text Mining and Dynamic Prediction in the Healthcare domains as a form of feature extraction and reduction.

We have introduced the use of the GD distribution in probabilistic topic modeling as a form to model a constraint set correlations. We have mentioned the advantages of the GD over the Dirichlet distribution, and its benefits, when compared with the estimation of the full covariance matrix in CTM, have been described.

The apparent constraints on covariances of the GD distribution result in modeling better sparse topic correlations in natural language documents, as our empirical validation indicates. This validation is based on empirical likelihood estimates and Information Retrieval based measures.

Due to the tree structure of the GD distribution, GD-LDA proves to be powerful in handling over-fitting with a large number of topics as its performance remains
fairly high even when the number of topics is increased. We have shown that the use of GD-LDA in ad-hoc IR increases the performance significantly, in contrast to earlier incorporations of topic models.

In addition, we have demonstrated that the running time of GD-LDA is comparable to LDA and less than other correlation-based methods such as CTM and PAM. This provides us a model computationally competitive and with better performance than these methods. Future work in this area includes providing a variant of the method that allow us to train the model in webscale. We can achieve this by training the model in a distributed form. We can train the model using the map-reduce framework by relaxing the sequential assumption of the Gibbs Sampling when fitting the model.

In the applications of Statistical Topic Models in Document Retrieval, we have presented two methods that incorporate topic models: one using relevance feedback and combining the topic mixture in the score function. Results in this application show that including the relevant and nonrelevant mixture of topics in the ranking formula improves the performance by reducing the document search space while adding context to the query.

We notice that performance of the results is highly dependent on the order the documents are retrieved. Future lines of work in this application include eliminating the ordering factor by maximizing the document relevance together with the minimization of the document variance. Another path is the incorporation of an update policy based on the nature of the labeled documents.

The second application covered in this thesis is based on the retrieval of health
care related documents using the patient query and related discharge summary. We have shown that the results obtained by including topic models in the retrieval of healthcare documents outperformed ontology-based approaches and standard based bag of words approaches.

In addition, we corroborated that using this technique, we could capture the document context more effectively. We have shown that Topic Models result in a more suitable unsupervised clustering technique when using a multisource document corpus due to the vocabulary heterogeneity. We also have shown that we could increase the performance of the model by including user’s context using the information obtained from the discharge summaries.

Further work in this application includes performing a Learning to Rank algorithm. Here we can combine the scores from the retrieval using only Topic Modeling and Topic Modeling combined with the discharge summaries with the clarity of the query as a weighting factor. Exploiting the correlation of the topics found in the discharge summaries as a method of query disambiguation is another research path to explore in this application.

Finally, we have proposed a method to model the latent health state of the patient using heterogeneous data dynamically. We obtain this data from the patient Electronic Medical Records. Using this latent state, we predict three different measures. These are the in-hospital probability of mortality inside the ICU using an aggregated set of features, the probability of patient readmission to the ICU during the next 30 days and the probability of system failure inside the ICU. We use the last measure to
estimate the probability of in-hospital mortality.

Our final goal is to provide tools that aid the doctors to process the different sources of data. We have shown that the use of an aggregated patient state, which combines current features with previously observed ones, allows us to predict the probability better the measures described before even if not new features are observed.

In this model, we treat the patient 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.

The results of the proposed model depend on the quality and quantity of the patient observed features. The more features values are observed for a given patient in his EMR and the more diverse patient pool data is, the more accurate prediction would be. The accurate prediction of both probabilities of mortality and patient readmission in the ICU would lead to an efficient allocation of resources that potentially could reduce the cost of the patient transfer and care.

In addition, our goal is the aid the medical staff to reduce the number of false alarms in the prediction by increasing the specificity of the model. This false alarm reduction allows the physicians to be available when a true alarm arrives.

We expanded the aggregated latent health state model to propose a framework to predict different subsystem failure and then combine them to predict the patient mortality. Our method provides a fully dynamic framework that takes into account future uncertainty by training the model using the complete patient path from admission to discharge/death with high performance.
Future research paths in this application include expanding the model to include different types of patients based on age and diseases. This is particularly important in patients with heart diseases which their behavior is different than other patient types.

The second research direction is to take advantage of the dynamic nature of the model to predict not only the presence of subsystem failure or increase in the patient mortality, but also to predict how likely the patient would be able to recover (no recovery, slow recovery with sequels or fast recovery with no sequels). We could also apply the same framework to predict the time delay between the patient transfer from the ICU to another area of the Hospital and the patient readmission to the ICU. By performing this estimation, we can evaluate how feasible is to discharge the patient to the ICU.
Bibliography


[7] Joel Barajas, Ram Akella, Marius Holtan, Jaimie Kwon, Aaron Flores, and Victor


[44] Le Gall JR, Lemeshow S, and Saulnier F. A new simplified acute physiology


[51] Kyung Soon Lee, W. Bruce Croft, and James Allan. A cluster-based resampling


[58] Peng Liu, Lei Lei, Junjie Yin, Wei Zhang, Wu Naijun, and Elia El-Darzi. Health-


[74] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth,


[80] Kent A. Spackman, Ph. D, Keith E. Campbell, Ph. D, Roger A. Cote, and D. Sc.


[87] Chunye Wang, Ram Akella, and Srikant Ramachandran. Hierarchical service


[94] Tzu-Tsung Wong. Parameter estimation for generalized dirichlet distributions
from the sample estimates of the first and the second moments of random variables.


[101] Youmin Zhang and X.R. Li. Fixed-interval smoothing algorithm based on singular

Appendix A

Generalized Dirichlet Distribution

A.1 First and Second Moments of the GD distribution

We derive the first and second moments of the GD distribution based on the tree representation of Eq 2.2. We write:

\[ \theta_1 = Z_1 \quad \theta_k = Z_k(1 - \theta_1 - \cdots - \theta_{k-1}) = Z_k \prod_{m=1}^{k-1} (1 - Z_m) \]  \hspace{1cm} (A.1)

Let \( S_k = \text{E}(Z_k) \), \( R_k = \text{E}(Z_k^2) \). Since \( Z_k \)'s are independent:

\[ \text{E}(\theta_k) = S_k \prod_{m=1}^{k-1} (1 - S_m), \quad \text{E}(\theta_k^2) = R_k \prod_{m=1}^{k-1} (1 - 2S_m + R_m) \]  \hspace{1cm} (A.2)

Similarly for the crossproducts where \( k < j \):

\[ \text{E}(\theta_k \theta_j) = E \left\{ Z_k \left[ \prod_{m=1}^{k-1} (1 - Z_m) \right] Z_j \left[ \prod_{m=1}^{j-1} (1 - Z_m) \right] \right\} \]
\[ = E \left\{ \left[ \prod_{m=1}^{k-1} (1 - Z_m)^2 \right] Z_k (1 - Z_k) \left[ \prod_{m=i+1}^{j-1} (1 - Z_m) \right] Z_j \right\} \]  \hspace{1cm} (A.3)
\[ = \left[ \prod_{m=1}^{k-1} (1 - 2S_m + R_m) \right] (S_k - R_k) \left[ \prod_{m=i+1}^{j-1} (1 - S_m) \right] S_j \]
Therefore, we have for the variance and covariance:

\[
\text{Var}(\theta_k) = R_k \left[ \prod_{m=1}^{k-1} (1 - 2S_m + R_m) \right] - S_k^2 \prod_{m=1}^{k-1} (1 - S_m)^2
\]

(A.4)

\[
\text{Cov}(\theta_k, \theta_j) = S_j \left[ \prod_{m=i+1}^{j-1} (1 - S_m) \right]
\]

\[
\{ (S_k - R_k) \prod_{m=1}^{k-1} (1 - 2S_m + R_m) - (S_k - S_k^2) \prod_{m=1}^{k-1} (1 - S_m)^2 \}
\]

(A.5)

\[
= \left[ E(\theta_j)/E(V_k^2) \right] \left[ S_k \text{Var}(V_k) - \text{Var}(\theta_k) \right]
\]

for \( k = 1, \ldots, K \) in Eq A.4, and for \( k = 1, \ldots, K - 1 \) and \( j = k + 1, \ldots, K \) in Eq A.5. We estimate \( S_k, R_k \) based on the independent Beta distributions of \( Z_k \). From the moment generating function for the Beta distribution we have:

\[
S_k = \frac{\alpha_k}{\alpha_k + \beta_k} \quad R_k = \frac{\alpha_k(\alpha_k + 1)}{(\alpha_k + \beta_k)(\alpha_k + \beta_k + 1)} \quad k = 1, \ldots, K
\]

(A.6)

where \( \alpha_K = 1, \beta_K = 0 \) [20]. Note that the variances and the expected value are not constrained.

### A.1.1 Covariance Properties of GD Distribution

Three key properties can be derived from Eqs A.4-A.6:

1. For \( j > k \), \( \text{Cov}(\theta_k, \theta_j) > 0 \) if and only if \( \text{Var}(V_k) > \text{Var}(\theta_k) \). Thus, \( \text{Cov}(\theta_k, \theta_j) > 0 \) if and only if the summation of the first \( k - 1 \) \( \theta \)’s varies more than \( \theta_k \)

2. For \( j > 1 \), \( \text{Cov}(\theta_1, \theta_j) = -\text{Var}(\theta_1)E(\theta_j)/E(1-\theta_1) < 0 \) since \( \theta_1 \) is at the root level of the tree

3. For \( j > k + 1 \), \( \text{Cov}(\theta_k, \theta_j) = \text{Cov}(\theta_k, \theta_{k+1})E(\theta_j)/E(\theta_{k+1}) \). Thus, \( \text{Cov}(\theta_k, \theta_j) \) at deeper levels of the tree will have the same sign as \( \text{Cov}(\theta_k, \theta_{k+1}) \)
The GD distribution models the $\text{Cov}(\theta_k, \theta_{k+1})$ for consecutive tree levels without constraints. The covariances for deeper levels in the tree $\text{Cov}(\theta_k, \theta_j), j > k + 1$ are constrained. Since $\theta_1$ is used as base category, it is always negatively correlated with the other categories. This is a typical constraint of the Logistic Normal distribution used in CTM[12]. The GD distribution constrains the sign of $\text{Cov}(\theta_k, \theta_j)$ to be the same as that of $\text{Cov}(\theta_k, \theta_{k+1})$ for $j > k + 1$. These constraints imply that probability of co-occurrence of these topics is very small.

A.2 Maximization of $\alpha$ and $\beta$ for GD

To estimate the parameters of the GD distribution, the log-likelihood of Eq. 2.6 is optimized using the Newton method.

$$L(\alpha, \beta) = \sum_{j=1}^{D} \sum_{k=1}^{K-1} \log \Gamma(\alpha_k + \beta_k) - \sum_{k=1}^{K-1} (\log \Gamma(\alpha_k) + \log \Gamma(\beta_k))$$

$$+ \sum_{k=1}^{K-1} \left( \log \Gamma(\alpha_k^j) + \log \Gamma(\beta_k^j) \right) - \sum_{k=1}^{K-1} \log \Gamma(\alpha_k^j + \beta_k^j)$$

(A.7)

By taking the derivative with respect to $\alpha, \beta$ we have:

$$\frac{\partial L(\alpha, \beta)}{\partial \alpha_k} = D\Psi(\alpha_k + \beta_k) - D\Psi(\alpha_k) + \sum_{j=1}^{D} \Psi(\alpha_k^j) - \sum_{j=1}^{D} \Psi(\alpha_k^j + \beta_k^j)$$

$$\frac{\partial L(\alpha, \beta)}{\partial \beta_k} = D\Psi(\alpha_k + \beta_k) - D\Psi(\beta_k) + \sum_{j=1}^{D} \Psi(\beta_k^j) - \sum_{j=1}^{D} \Psi(\alpha_k^j + \beta_k^j)$$

(A.8)

Recall that $\alpha_k^j = \alpha_k + \sum_{i=j,k} N_{j,k}, \beta_k^j = \beta_k + \sum_{i=j+1} N_{j,k+1} + \ldots + N_{j,K}$. Notice that the derivative of $L(\alpha, \beta)$ with respect to $\alpha_k$ just depends on $\alpha_k$ and $\beta_k$ as opposed to the Dirichlet
distribution. For the second derivative we have:

\[
\frac{\partial^2 L(\alpha, \beta)}{(\partial \alpha_k)^2} = D\Psi'(\alpha_k + \beta_k) - D\Psi'(\alpha_k) + \sum_{j=1}^{D} \Psi'(\alpha_k^j) - \sum_{j=1}^{D} \Psi'(\alpha_k^j + \beta_k^j)
\]

\[
\frac{\partial^2 L(\alpha, \beta)}{(\partial \beta_k)^2} = D\Psi'(\alpha_k + \beta_k) - D\Psi'(\beta_k) + \sum_{j=1}^{D} \Psi'(\beta_k^j) - \sum_{j=1}^{D} \Psi'(\alpha_k^j + \beta_k^j)
\]

\[
\frac{\partial^2 L(\alpha, \beta)}{\partial \alpha_k \partial \beta_k} = D\Psi'(\alpha_k + \beta_k)
\]

\[
\frac{\partial^2 L(\alpha, \beta)}{\partial \alpha_l \partial \beta_k} = 0, \quad \frac{\partial^2 L(\alpha, \beta)}{\partial \beta_k \partial \beta_l} = 0 \text{ for } l \neq k
\]

Therefore, the Hessian matrix can be written as:

\[
H(\alpha, \beta) = \text{block-diag} \left[ H_1(\alpha_1, \beta_1), \ldots, H_{K-1}(\alpha_{K-1}, \beta_{K-1}) \right]
\]

where \(H_k\) is the Hessian matrix for \(\alpha_k, \beta_k\). Following a similar logic as in the case of a Dirichlet distribution described in Eqs (56-60) of [67], we have the following Newton iteration:

\[
\alpha_k^{\text{new}} = \alpha_k - (H_k^{-1} g_k)_1, \quad \beta_k^{\text{new}} = \beta_k - (H_k^{-1} g_k)_2
\]

\[
q_{11}^k = \sum_{j=1}^{D} \Psi'(\alpha_k^j) - D\Psi'(\alpha_k), \quad q_{22}^k = \sum_{j=1}^{D} \Psi'(\beta_k^j) - D\Psi'(\beta_k)
\]

\[
a_k = \frac{g_{11}^k / q_{11}^k + g_{22}^k / q_{22}^k}{b_k^{-1} + (q_{11}^k)^{-1} + (q_{22}^k)^{-1}}, \quad b_k = D\Psi'(\alpha_k + \beta_k) - \sum_{j=1}^{D} \Psi'(\alpha_k^j + \beta_k^j)
\]

\[
(H_k^{-1} g_k)_l = \frac{g_{lk}^k - a_k}{q_{ll}^k} \quad \text{for } l = 1, 2
\]

where \(g_{11}^k = dL(\alpha, \beta)/d\alpha_k\) and \(g_{22}^k = dL(\alpha, \beta)/d\beta_k\) from Eq A.8.