A Classification Tool for Predictive Data Analysis in Healthcare

Mason Lemoyne Victors
Brigham Young University - Provo

A Classification Tool for Predictive Data Analysis in Healthcare

Mason Victors

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mathematics

Brigham Young University

March 2013

ABSTRACT

A Classification Tool for Predictive Data Analysis in Healthcare

Mason Victors
Department of Mathematics, BYU
Master of Science

Hidden Markov Models (HMMs) have seen widespread use in a variety of applications ranging from speech recognition to gene prediction. While developed over forty years ago, they remain a standard tool for sequential data analysis. More recently, Latent Dirichlet Allocation (LDA) was developed and soon gained widespread popularity as a powerful topic analysis tool for text corpora. We thoroughly develop LDA and a generalization of HMMs and demonstrate the conjunctive use of both methods in predictive data analysis for health care problems.

While these two tools (LDA and HMM) have been used in conjunction previously, we use LDA in a new way to reduce the dimensionality involved in the training of HMMs. With both LDA and our extension of HMM, we train classifiers to predict development of Chronic Kidney Disease (CKD) in the near future.

Keywords: predictive data analysis, Hidden Markov Models, Latent Dirichlet Allocation, health care, convex analysis, Markov chains, Expectation Maximization, Gibbs sampling, classification tree, random forest

Acknowledgments

My first thanks go to my wife Sarah, who has been there to share in my excitement over successes and disappointments over setbacks. Her constant support and encouragement throughout my graduate studies have made this work possible. I am also grateful for the many students with whom it has been my privilege to work during the past two years. Particular mention should be made of Ryan Grout and Matthew Webb, who have been extremely helpful in answering frequent questions about programming and in sounding out many of the general ideas presented here. I would also like to deeply thank BYU Graduate Studies for providing the research fellowship that has allowed me to spend so much time on this project.

Finally, I must acknowledge the efforts of my advisor Jeffrey Humpherys. His guidance and counsel have been invaluable to me, and are not limited to this academic work. I am especially grateful for his strong encouragement to stay at BYU for another degree as well as the countless hours spent teaching me to recognize the most important mathematical tools and models.

BRIGHAM YOUNG UNIVERSITY

SIGNATURE PAGE

of a thesis submitted by

Mason Victors

The thesis of Mason Victors is acceptable in its final form including (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory and ready for submission.

Date	Jeffrey Humpherys, Chair
Date	Shane Reese
Date	Eric Ringger
Date	Stephen Humphries, Director of Graduate Studies
Date	Thomas W. Sederberg, Associate Dean College of Physical and Mathematical Sciences

Contents

List of Tables

List of Figures

Chapter 1. Introduction

As researchers and scientists throughout the world gather and generate more and more data in their academic and industrial pursuits, analysis of that data is becoming increasingly more difficult in scope and yet in greater demand than ever. Whether it be for medical purposes in improving gene prediction in DNA sequences or the timely analysis of email correspondence of potential terrorists acquired by the U.S. Intelligence Community, pattern recognition is in constant demand by professionals and policymakers everywhere, and thus research to develop additional methods and improve existing ones should be a priority of the entire scientific community.

In particular, the realm of health care has been designated as ripe for breakthroughs in data analysis, yielding upwards of $300 billion in value each year [1]. The ability to accurately predict near-future development of a chronic disease such as Chronic Kidney Disease (CKD) would lead to a much higher quality of life for tens of thousands of Americans each year, and provide immense savings in health care spending as well. With this motivation, we developed new techniques to build a classifier for early disease diagnosis. While the methods used are standard tools in data analysis, their conjunctive use and application is somewhat unique.

In Chapter 2 we provide a thorough introduction to Markov chains and convex analysis. In Chapter 3 we show the details of two landmark statistical algorithms: Gibbs sampling and Expectation-Maximization. We also include sections on classification trees, random forests, and how to measure the success of a binary classifier, as these will ultimately be used in our analysis in Chapter 6. In Chapters 4 and5 we use these algorithms to thoroughly develop the inference procedures of LDA and our generalization of HMMs. In Chapter 6 we present the use of these two tools in a classifier to predict the development of CKD, and we conclude this work in Chapter 7.

To keep this work as self-contained as possible, we provide numerous proofs of well-established (but nontrivial) mathematical results, while still citing outside sources for furthur reading and understanding. We omit more extensive proofs and provide references to the appropriate material. We hope this will make this work more approachable by those with little background in the processes used.

Chapter 2. Background

In this chapter we start with a thorough development of key aspects of Markov chains and then delve into convex analysis.

2.1 Markov Chains

Many mathematical and statistical methods involve a Bayesian sampling procedure. These procedures are collectively referred to as Markov Chain Monte Carlo (MCMC) methods, and they hinge on the theory of Markov chains. The theory justifying these sampling procedures is worth learning well, so as to ensure that any MCMC method is used appropriately. We include the necessary pieces of this theory for completion. These theorems and proofs, as well as a more thorough discussion of Markov chains, can be found in [2]. We must begin with several definitions.

Definition 2.1. A sequence of random variables $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots\}$ assuming values from some state space Q is called a *Markov chain* if it satisfies the following condition:

$$\mathbb{P}(\mathbf{x}_{t+1} = s | \mathbf{x}_1 = s_1, \mathbf{x}_2 = s_2, \cdots, \mathbf{x}_t = s_t) = \mathbb{P}(\mathbf{x}_{t+1} = s | \mathbf{x}_t = s_t). \tag{2.1}$$

In other words, as the random process transitions from its current state to the next, this transition is dependent only on the current state, in other words, given the current state, it is independent of all prior states.

Such a process is called *homogeneous* if for each fixed $q, r \in Q$ we have

$$\mathbb{P}(\mathbf{x}_{t+1} = q | \mathbf{x}_t = r) = \mathbb{P}(\mathbf{x}_2 = q | \mathbf{x}_1 = r) \text{ for all } t \in \mathbb{N}. \tag{2.2}$$

In this case, we may then fix $p_{xy} = \mathbb{P}(\mathbf{x}_{t+1} = x|\mathbf{x}_t = y)$, as the value of this probability is time-independent. If $Q = \{s_1, \cdots, s_N\}$ is a finite state space, these transitions can be modeled by a single $N \times N$ column-stochastic matrix $A = (a_{ij})$, where

$$a_{ij} = \mathbb{P}(\mathbf{x}_{t+1} = j|\mathbf{x}_t = i), \tag{2.3}$$

where throughout we identify s_i with i. Thus a_{ij} is the probability of transitioning to state j, given that the current state is i, and we refer to A throughout as the transition probability matrix. We let A_i denote the i^{th} row of A. Throughout we also assume that all Markov chains are homogeneous with finite state space Q. A Markov chain can be graphically represented by the following diagram:

$$\begin{tikzcd} \mathbf{x}_1 \arrow[r] & \mathbf{x}_2 \arrow[r] & \cdots \arrow[r] & \mathbf{x}_{t-1} \arrow[r] & \mathbf{x}_t \arrow[r] & \mathbf{x}_{t+1} \arrow[r] & \cdots \end{tikzcd}$$

Throughout we also write $a_{ij}(t) = \mathbb{P}(\mathbf{x}_t = j|\mathbf{x}_0 = i)$. This is called the t^{th} *step transition probability*, and there is a convenient way to compute this via matrix multiplication.

Theorem 2.2. *Let Q be a state space with size N and A the transition matrix for a Markov chain. Then $A(t) = A^t$, that is, $a_{ij}(t) = (A^t)_{ij}$ for each $1 \leq i, j \leq N$.*

Proof. Note that $A(1) = A = A^1$. Now suppose that for some t we have $A(t) = A^t$. Then

$$\begin{aligned}
a_{ij}(t+1) &= \mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_0 = i) \\
&= \sum_{k=1}^{N} \mathbb{P}(\mathbf{x}_{t+1} = j, \mathbf{x}_t = k | \mathbf{x}_0 = i) \\
&= \sum_{k=1}^{N} \mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_t = k, \mathbf{x}_0 = i)\mathbb{P}(\mathbf{x}_t = k | \mathbf{x}_0 = i) \\
&= \sum_{k=1}^{N} a_{ik}(t) a_{kj} \\
&= \sum_{k=1}^{N} (A^t)_{ik} a_{kj} \\
&= (A^t A)_{ij} \\
&= (A^{t+1})_{ij}
\end{aligned}$$

□

We require a way to distinguish between different classes of states: those which will probabilistically recur, and those which will not.

Definition 2.3. A state $i \in Q$ is called *recurrent* if

$$\mathbb{P}(\mathbf{x}_t = i \text{ for some } t \geq 1 | \mathbf{x}_0 = i) = 1.$$

Otherwise it is called *transient*. We also define

$$f_{ij}(t) = \mathbb{P}(\mathbf{x}_1 \neq j, \cdots, \mathbf{x}_{t-1} \neq j, \mathbf{x}_t = j | \mathbf{x}_0 = i), \quad f_{ij} = \sum_{t=1}^{\infty} f_{ij}(t).$$

Note that $f_{ij} \leq 1$ for each pair i, j. For each pair of states i, j, we can define generating functions

$$P_{ij}(s) = \sum_{t=0}^{\infty} s^t a_{ij}(t), \quad F_{ij}(s) = \sum_{t=0}^{\infty} s^t f_{ij}(t).$$

Here we define $f_{ij}(0) = 0$ for all $i, j \in Q$ and always assume $|s| < 1$ so that $P_{ij}(s)$ and $F_{ij}(s)$ converge.

Lemma 2.4. *A state $i \in Q$ is recurrent if and only if $f_{ii} = 1$.*

Proof. This follows immediately from the definition of $f_{ii}(t)$. □

The following theorem makes a claim that seems intuitive: given any starting state, the probability of being at a specific transient state at time t approaches 0 as t gets large. That is, as time passes it becomes less and less likely that the chain is in a transient state.

Theorem 2.5. *If j is transient, then $\lim_{t\to\infty} a_{ij}(t) = 0$ for all $i \in Q$.*

Proof. Fix $i, j \in Q$, j transient and let $B_n = \{\mathbf{x}_1 \neq j, \cdots, \mathbf{x}_{n-1} \neq j, \mathbf{x}_n = j\}$. Note that

$$\mathbb{P}(\mathbf{x}_t = j|\mathbf{x}_0 = i) = \sum_{n=1}^{t} \mathbb{P}(\mathbf{x}_t = j, B_n|\mathbf{x}_0 = i)$$

since the events B_n partitition the event $\{\mathbf{x}_t = j\}$ as n ranges from 1 to t. But

$$\begin{aligned}\mathbb{P}(\mathbf{x}_t = j, B_n|\mathbf{x}_0 = i) &= \mathbb{P}(\mathbf{x}_t = j|\mathbf{x}_0 = i, B_n)\mathbb{P}(B_n|\mathbf{x}_0 = i)\\ &= \mathbb{P}(\mathbf{x}_t = j|\mathbf{x}_n = j)\mathbb{P}(B_n|\mathbf{x}_0 = i)\\ &= a_{ij}(t-n)f_{ij}(n).\end{aligned}$$

Thus $a_{ij}(t) = \sum_{n=0}^{t} a_{ij}(t-n)f_{ij}(n)$ since $f_{ij}(0) = 0$ for any $i, j \in Q$. But then

$$\begin{aligned}P_{ij}(s) - \mathbb{1}_{[i=j]} &= \sum_{t=1}^{\infty} s^t a_{ij}(t)\\ &= \sum_{t=0}^{\infty} s^t \sum_{n=0}^{t} a_{ij}(t-n)f_{ij}(n)\\ &= \sum_{n=0}^{\infty} s^n f_{ij}(n) \sum_{t=n}^{\infty} s^{t-n} a_{ij}(t-n)\\ &= F_{ij}(s)P_{ij}(s)\end{aligned}$$

where $|s| < 1$. For such s, we have $P_{jj}(s) = \frac{1}{1-F_{jj}(s)}$. We have by Abel's Theorem [2],

$$\lim_{s\uparrow 1} P_{jj}(s) = \sum_{t=0}^{\infty} a_{jj}(t), \quad \lim_{s\uparrow 1} F_{jj}(s) = \sum_{t=0}^{\infty} f_{jj}(t) = f_{jj}.$$

Thus

$$\sum_{t=0}^{\infty} a_{jj}(t) = \infty \Leftrightarrow f_{jj} = 1 \Leftrightarrow j \text{ is recurrent.}$$

Suppose j is transient. Then $f_{jj} < 1$, so $\sum_{t=0}^{\infty} a_{jj}(t) < \infty$. Also, if $i \neq j$, then we have $P_{ij}(s) = F_{ij}(s)P_{jj}(s)$, so

$$\begin{aligned} \sum_{t=0}^{\infty} a_{ij}(t) &= f_{ij} \sum_{t=0}^{\infty} a_{jj}(t) \\ &\leq \sum_{t=0}^{\infty} a_{jj}(t) \\ &< \infty \end{aligned}$$

But then this series converges, so $\lim_{t\to\infty} a_{ij}(t) = 0$ for any $i, j \in Q$ with j transient. $\square$

Definition 2.6. Let $T_i = \min\{t \geq 1 : \mathbf{x}_t = i, \mathbf{x}_0 = i\}$. Then the *mean recurrence time* μ_i is defined as

$$\mu_i = \mathbb{E}(T_i).$$

While interesting itself, the mean recurrence time for a state plays a significant role in the Gibbs sampling process, as we will soon see.

Definition 2.7. A recurrent state i is called *non-null recurrent* if $\mu_i < \infty$ but is called *null recurrent* if $\mu_i = \infty$. Note that if s is transient, then we always have $\mu_i = \infty$.

Definition 2.8. A Markov chain $\mathbf{X}$ over a state space Q is *irreducible* if for each $i, j \in Q$ we have that

$$a_{ij}(m) > 0$$

for some $m > 0$. In this case we say i *communicates with* j.

Lemma 2.9. *Let $i, j \in Q$ and suppose i and j communicate with each other. Then i is transient if and only if j is transient.*

Proof. Since i and j communicate with each other, then there exist $m, n \in \mathbb{N}$ such that $a_{ij}(m) > 0$ and $a_{ji}(n) > 0$, and let $\alpha = a_{ij}(m)a_{ji}(n) > 0$. Let

$$T = \{\mathbf{X} | \mathbf{x}_0 = i, \mathbf{x}_{m+n+r} = i, r \geq 0\},$$

the set of all possible Markov chains starting at i and ending at i after $m+n+r$ transitions. If a chain starts at i and transitions to j after m iterations, then returns to j after another r iterations, and finally back to i after n more transitions, then this chain is in T. Thus

$$a_{ii}(m+r+n) \geq a_{ij}(m)a_{jj}(r)a_{ji}(n) = \alpha a_{jj}(r)$$

where $r \geq 0$. By this work and that of Theorem 2.5, we have the following chain of implications:

$$i \text{ is transient} \Rightarrow \sum_{r=0}^{\infty} a_{ii}(r) < \infty \Rightarrow \sum_{r=0}^{\infty} a_{jj}(r) < \infty \Rightarrow j \text{ is transient.}$$

The converse follows from renaming i and j. □

We are now ready to discuss the meat of elementary Markov chain theory. The following theorems can alternatively be developed by consideration of the spectrum of the transition matrix (its eigenvalues and eigenvectors), ultimately hinging on the Perron-Frobenius Theorem.

Definition 2.10. A distribution $\boldsymbol{\pi}$ over the state space Q is an *invariant distribution* if

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T A. \tag{2.4}$$

which means that $\boldsymbol{\pi}_j = \sum_{i=1}^{N} \boldsymbol{\pi}_i a_{ij}$ for all j. Note that $\boldsymbol{\pi}$ is a vector such that $\boldsymbol{\pi}_i \geq 0$ for all i and $\sum_i \boldsymbol{\pi}_i = 1$.

Lemma 2.11. *Let $\boldsymbol{\pi}$ be an invariant distribution for a Markov chain with state space Q and transition matrix A. Then $\boldsymbol{\pi}^T = \boldsymbol{\pi}^T A^t$ for all $t \geq 0$.*

Proof. This follows immediately from the definition of an invariant distribution and a simple inductive argument on t. □

Theorem 2.12. *If $\mathbf{X}$ is an irreducible Markov chain with an invariant distribution $\boldsymbol{\pi}$, then this invariant distribution is unique.*

Proof. Suppose some state $j \in Q$ is transient. Then since $\mathbf{X}$ is irreducible, j communicates with all other states, so every state is transient. But then $\lim_{t\to\infty} a_{ij}(t) = 0$ for all $i, j \in Q$ by Theorem 2.5. Since $\boldsymbol{\pi}$ is invariant, we have

$$\boldsymbol{\pi}_j = \sum_{i=1}^{N} \boldsymbol{\pi}_i a_{ij}(t)$$

for any j and t. But then

$$\begin{aligned}\boldsymbol{\pi}_j &= \lim_{t\to\infty} \sum_{i=1}^{N} \boldsymbol{\pi}_i a_{ij}(t) \\ &= \sum_{i=1}^{N} \boldsymbol{\pi}_i \lim_{t\to\infty} a_{ij}(t) = 0\end{aligned}$$

for all $j \in Q$, in which case $\boldsymbol{\pi}$ is not a distribution, a contradiction. Thus all states are recurrent.

Suppose now that $\boldsymbol{\pi}_j = 0$ for some $j \in Q$. Then we have

$$0 = \boldsymbol{\pi}_j = \sum_{i=1}^{N} \boldsymbol{\pi}_i a_{ij}(t) \geq \boldsymbol{\pi}_i a_{ij}(t)$$

for any $i \in Q$ and t. But then since i and j communicate, there is some t such that $a_{ij}(t) > 0$. But then $\boldsymbol{\pi}_i = 0$ for all i, a contradiction to the stochasticity of $\boldsymbol{\pi}$. Thus $\boldsymbol{\pi}_j > 0$ for all $j \in Q$.

Let's suppose now that $\mathbf{x}_0$ has $\boldsymbol{\pi}$ as its distribution, and fix $j \in Q$. Then

$$\begin{aligned}
\boldsymbol{\pi}_j \mu_j &= \boldsymbol{\pi}_j \mathbb{E}(T_j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) \sum_{t=1}^{\infty} t\mathbb{P}(T_j = t | \mathbf{x}_0 = j) \\
&= \sum_{t=1}^{\infty} \mathbb{P}(T_j \geq t | \mathbf{x}_0 = j)\mathbb{P}(\mathbf{x}_0 = j) \\
&= \sum_{t=1}^{\infty} \mathbb{P}(T_j \geq t, \mathbf{x}_0 = j) \\
&= \mathbb{P}(T_j \geq 1, \mathbf{x}_0 = j) + \sum_{t=2}^{\infty} \mathbb{P}(T_j \geq t, \mathbf{x}_0 = j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \sum_{t=2}^{\infty} \mathbb{P}(\mathbf{x}_0 = j, \mathbf{x}_1 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \sum_{t=2}^{\infty} \mathbb{P}(\mathbf{x}_1 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) - \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \sum_{t=2}^{\infty} \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-2} \neq j) - \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \lim_{n\to\infty} \sum_{t=2}^{n} \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-2} \neq j) - \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \lim_{t\to\infty} \mathbb{P}(\mathbf{x}_0 \neq j) - \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= \mathbb{P}(\mathbf{x}_0 = j) + \mathbb{P}(\mathbf{x}_0 \neq j) - \lim_{t\to\infty} \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= 1 - \lim_{t\to\infty} \mathbb{P}(\mathbf{x}_0 \neq j, \cdots, \mathbf{x}_{t-1} \neq j) \\
&= 1
\end{aligned}$$

since j is recurrent. Since $\boldsymbol{\pi}_j > 0$ for all $j \in Q$, we have that $\mu_j = \frac{1}{\boldsymbol{\pi}_j} < \infty$, so $\boldsymbol{\pi}_j = \frac{1}{\mu_j}$, which is unique. □

Definition 2.13. A Markov chain $\mathbf{X}$ is called *aperiodic* if $\gcd\{t : a_{ii}(t) > 0\} = 1$ for all $i \in Q$.

Theorem 2.14. *If $\mathbf{X}$ is an irreducible, aperiodic Markov chain with an invariant distribution $\boldsymbol{\pi}$, then $\lim_{t\to\infty} a_{ij}(t) = \boldsymbol{\pi}_j$ for all $j \in Q$.*

Proof. The proof of this theorem is beyond the scope of this work. See [2]. □

The above theorem is a major result in the theory of Markov chains. It shows for a certain class of Markov chains, the state space distribution at time t approaches a unique invariant distribution $\boldsymbol{\pi}$ as t gets large, *no matter the initial distribution over the state space*. This is one factor which ultimately enables Gibbs sampling (and subsequently our implementation of Latent Dirichlet Allocation) to work.

Definition 2.15. A homogeneous Markov chain $\mathbf{X}$ is called *reversible* if there exists a probability distribution $\boldsymbol{\pi}$ over the state space Q such that

$$\boldsymbol{\pi}_i a_{ij} = \boldsymbol{\pi}_j a_{ji} \tag{2.5}$$

for all $i, j \in Q$, where $\boldsymbol{\pi}_i$ is the value of the probability density function of $\boldsymbol{\pi}$ at $i \in Q$ and a_{ij} is as explained above.

The following Lemma will be of great use in the development of Gibbs sampling:

Lemma 2.16. *If a homogeneous Markov chain* $\mathbf{X}$ *over a finite state space is reversible, then the given probability distribution* $\boldsymbol{\pi}$ *with respect to which* X *is reversible, is also an invariant distribution of* $\mathbf{X}$.

Proof. We show that $\sum_{i=1}^{n} a_{ij}\boldsymbol{\pi}_i = \boldsymbol{\pi}_j$ for arbitrary $j \in Q$. By reversibility,

$$\boldsymbol{\pi}_i a_{ij} = \boldsymbol{\pi}_j a_{ji}.$$

Summing over i, we have

$$\begin{aligned}\sum_{i=1}^{N} \boldsymbol{\pi}_i a_{ij} &= \sum_{i=1}^{N} \boldsymbol{\pi}_j a_{ji} && (2.6)\\ &= \boldsymbol{\pi}_j \sum_{i=1}^{N} a_{ji} && (2.7)\\ &= \boldsymbol{\pi}_j && (2.8)\end{aligned}$$

where the final step follows from the fact that A is row stochastic. □

2.2 Convex Analysis

Convex analysis provides the theory behind many optimization problems. Considering that Expectation-Maximization is an optimization problem itself (hence the name), it is fitting that we provide an introduction to convex analysis to understand why EM works as it does. Similar and more intensive approaches to convex analysis may be found in [3] and [4], respectively.

Definition 2.17. Let $I = [a, b] \subseteq \mathbb{R}$ and let $f : I \to \mathbb{R}$. We say f is *convex* on I if

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

for every $x_1, x_2 \in I$ and $\lambda \in [0, 1]$. We say f is *strictly convex* on I if the inequality is strict for all $\lambda \in (0, 1)$.

Before we establish a powerful fact about convex functions, we need the following Lemma.

Lemma 2.18. *Let f be differentiable on I. If*

$$f(x) + f'(x)(y - x) \leq f(y)$$

for all $x, y \in I$, then f is convex on I.

Proof. Suppose $f(x) + f'(x)(y - x) \leq f(y)$ for all $x, y \in I$. Let $u, v \in I$ and $\lambda \in (0, 1)$. Let $w = \lambda u + (1 - \lambda)v$. Then $w \in I$ and we have

$$v - w = -\lambda(u - v)$$

$$u - w = (1 - \lambda)(u - v)$$

so

$$v - w = -\frac{\lambda}{1-\lambda}(u-w).$$

Then by our hypothesis,

$$\begin{aligned} f(w) + f'(w)(u-w) &\leq f(u) \\ f(w) - \left(\frac{\lambda}{1-\lambda}\right) f'(w)(u-w) &\leq f(v). \end{aligned}$$

Multiplying by λ and $1-\lambda$ appropriately and adding, this yields

$$f(w) \leq \lambda f(u) + (1-\lambda) f(v).$$

Since $w = \lambda u + (1-\lambda)v$, we have now shown that f is convex. □

Recall the well established Taylor's Theorem from calculus:

Theorem 2.19. *Let $I = [a,b] \subseteq \mathbb{R}$ be an interval and let f be a twice differentiable function on I. Let $x, y \in I$, not equal. Then there exists $z \in [x,y]$ such that*

$$f(x) = f(y) + f'(y)(x-y) + \frac{1}{2} f''(z)(x-y)^2.$$

We can now show a powerful connection between convexity and the second derivative of a function f:

Theorem 2.20. *If f is twice differentiable on an interval $I = [a,b]$ and $f''(x) \geq 0$ for all $x \in I$, then f is convex on I.*

Proof. Let $x, y \in I$. Then by Taylor's Theorem, there is some $z \in I$ such that

$$f(x) = f(y) + f'(y)(x-y) + \frac{1}{2} f''(z)(x-y)^2.$$

Since $f''(z) \geq 0$ and $(x-y)^2 \geq 0$, then

$$f(x) \geq f(y) + f'(y)(x-y).$$

By our lemma, we conclude that f is convex on I. $\square$

This gives rise to the following Corollary:

Corollary 2.21. $-\ln(x)$ *is convex on the interval* $(0, \infty)$.

Proof. Let $f(x) = -\ln(x)$. Then $f'(x) = -\frac{1}{x}$ and $f''(x) = \frac{1}{x^2} \geq 0$ for all $x \in (0, \infty)$. By the Theorem, $-\ln(x)$ is convex. $\square$

We now give one of the most important inequalities of all of mathematics: Jensen's Inequality. Many important inequalities are simple cases of Jensen's Inequality, including Young's, Holder's, Minkowski's, and the Arithmetic-Geometric Mean inequalities.

Theorem 2.22. *Jensen's Inequality Let* f *be a convex function defined on an interval* $I \subseteq \mathbb{R}$. *Let* $x_1, \cdots, x_n \in I$ *and* $\lambda_1, \cdots, \lambda_n \geq 0$ *such that* $\sum_{i=1}^n \lambda_i = 1$. *Then*

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i).$$

Proof. Let $n = 1$. Then we have $f(x_1) \leq f(x_1)$ trivially. If $n = 2$, then this is the definition of a convex function. Suppose the inequality holds for some n. Then

$$\begin{aligned}
f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) &= f\left(\sum_{i=1}^{n}\lambda_i x_i + \lambda_{n+1}x_{n+1}\right)\\
&= f\left((1-\lambda_{n+1})\frac{1}{1-\lambda_{n+1}}\sum_{i=1}^{n}\lambda_i x_i + \lambda_{n+1}x_{n+1}\right)\\
&\le (1-\lambda_{n+1})f\left(\frac{1}{1-\lambda_{n+1}}\sum_{i=1}^{n}\lambda_i x_i\right) + \lambda_{n+1}f(x_{n+1})\\
&= (1-\lambda_{n+1})f\left(\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}x_i\right) + \lambda_{n+1}f(x_{n+1})\\
&\le (1-\lambda_{n+1})\sum_{i=1}^{n}\frac{\lambda_i}{1-\lambda_{n+1}}f(x_i) + \lambda_{n+1}f(x_{n+1})\\
&= \sum_{i=1}^{n+1}\lambda_i f(x_i).
\end{aligned}$$

□

We follow this with another corollary:

Corollary 2.23. *Let* $x_1, \cdots, x_n > 0$ *and* $\lambda_1, \cdots, \lambda_n \ge 0$ *such that* $\sum_{i=1}^{n}\lambda_i = 1$. *Then*

$$\ln\left(\sum_{i=1}^{n}\lambda_i x_i\right) \ge \sum_{i=1}^{n}\lambda_i \ln x_i.$$

Proof. Since $-\ln x$ is a convex function, then by Jensen's Inequality,

$$-\ln\left(\sum_{i=1}^{n}\lambda_i x_i\right) \le \sum_{i=1}^{n}\lambda_i(-\ln x_i).$$

Multiplying by -1, we get the desired inequality. □

We conclude our section on convexity with one final theorem which will prove useful later:

Theorem 2.24. *a) If $f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})$ are convex functions over a convex set $C \subseteq \mathbb{R}^n$, then*

$$g(\mathbf{x}) = \sum_{i=1}^{k} f_i(\mathbf{x})$$

is also convex.

b) If $f(\mathbf{x})$ is a convex function over a convex set $C \subseteq \mathbb{R}^n$, and if $c \in \mathbb{R}$ is nonnegative, then $cf(\mathbf{x})$ is convex.

c) Linear functions are convex.

Proof. Throughout the proof let $\mathbf{x}, \mathbf{y} \in C$ and let $\lambda \in (0, 1)$.

a) From the convexity of each f_i, we have

$$\begin{aligned} g(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) &= \sum_{i=1}^{k} f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \\ &\leq \sum_{i=1}^{k} \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) \\ &= \lambda \sum_{i=1}^{k} f(\mathbf{x}) + (1-\lambda) \sum_{i=1}^{k} f(\mathbf{y}) \\ &= \lambda g(\mathbf{x}) + (1-\lambda) g(\mathbf{y}) \end{aligned}$$

Thus $g(\mathbf{x})$ is convex.

b) If $c = 0$, then this is trivial, since

$$cf(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) = 0 = \lambda(0) + (1-\lambda)(0) = \lambda(cf(\mathbf{x})) + (1-\lambda)(cf(\mathbf{y})).$$

Suppose then that $c > 0$. Then since $f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y})$, multiplying by a positive constant preserves the inequality, yielding

$$\begin{aligned} cf(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) &\leq c(\lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y})) \\ &= \lambda cf(\mathbf{x}) + (1-\lambda) cf(\mathbf{y}) \end{aligned}$$

Thus $cf(\mathbf{x})$ is convex.

c) A linear function in $\mathbb{R}^n$ must be of the following form: $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + b$ for some $\mathbf{c} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Then

$$\begin{aligned} f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) &= \mathbf{c}^T(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) + b \\ &= \lambda\mathbf{c}^T\mathbf{x} + (1-\lambda)\mathbf{c}^T\mathbf{y} + b \\ &= \lambda(\mathbf{c}^T\mathbf{x} + b) + (1-\lambda)(\mathbf{c}^T\mathbf{y} + b) \\ &= \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y}) \end{aligned}$$

Thus $f(\mathbf{x})$ is convex. □

Chapter 3. Statistical Learning Algorithms

The main tools we use in this work depend heavily on two important statistical methods developed in the 20th century. We have thus far striven to develop the mathematical underpinnings of these algorithms, and now we present them in their entirety for the reader's benefit. More comprehensive/alternative introductions can be found in [5], [6], and [3], but for self-containment, we have provided all necessary mathematical details to satisfy the reader's desire to understand *why* they work. We also provide an introduction to two classification tools and a method to measure the effectiveness of a binary classifier.

3.1 Gibbs Sampling

In statistics, Gibbs sampling is an (MCMC) method used for approximating samples from a specified multivariate distribution $\boldsymbol{\pi}$ where direct sampling is infeasible, but where sampling from conditional distributions is simple. These approximated samples can then be used to estimate the joint distribution. Let S be a finite state space, and let $V \in \mathbb{N}$. Then $\Theta = S^V$

is also a finite state space. Let $\mathbf{X}$ be a random vector with range Θ, and let $\boldsymbol{\pi}$ be its joint mass function, i.e.

$$\boldsymbol{\pi}(s_1, \cdots, s_V) = \mathbb{P}(x_1 = s_1, \cdots, x_V = s_V).$$

Suppose that it is difficult to sample from $\boldsymbol{\pi}$, but that $\boldsymbol{\pi}(x_i | x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_V)$ is easy to sample from. Given values for an initial random vector $\mathbf{X}$, Gibbs sampling procedes as follows:

(i) Pick a random index $1 \leq i \leq V$.

(ii) Draw $x \sim \boldsymbol{\pi}(x_i | x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_V)$.

(iii) Fix $x_i = x$.

(iv) Repeat.

This procedure creates a reversible Markov chain over the finite state space Θ with invariant distribution $\boldsymbol{\pi}$.

Theorem 3.1. *Let $\boldsymbol{\pi}(x_1, \cdots, x_V)$ be a distribution from which we wish to sample and let Θ denote the sample space. Given an initial random vector $\mathbf{X} \in \Theta$, consider the method where we repeat the following three steps:*

(i) Pick a random index $1 \leq i \leq V$.

(ii) Pick a new value $x \sim \boldsymbol{\pi}(x_i | x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_V)$.

(iii) Fix $x_i = x$.

This process creates a reversible Markov chain on the sample space with invariant distribution $\boldsymbol{\pi}$.

Proof. Define a relation $\mathbf{X} \sim_j \mathbf{Y}$ if $x_i = y_i$ for all $i \neq j$, and note that this is an equivalence relation. With this process the transition probabilities from one sample $\mathbf{X} \in \Theta$ to $\mathbf{Y} \in \Theta$ are

$$q_{XY} = \begin{cases} \frac{1}{V} \frac{\boldsymbol{\pi}(\mathbf{Y})}{\sum\limits_{\substack{\mathbf{Z}\in\Theta \\ \mathbf{Z}\sim_j\mathbf{X}}} \boldsymbol{\pi}(\mathbf{Z})}, & \text{if } \mathbf{X} \sim_j \mathbf{Y}. \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$\boldsymbol{\pi}(\mathbf{X}) q_{XY} = \frac{1}{V} \frac{\boldsymbol{\pi}(\mathbf{X})\boldsymbol{\pi}(\mathbf{Y})}{\sum\limits_{\substack{\mathbf{Z}\in\Theta \\ \mathbf{Z}\sim_j\mathbf{X}}} \boldsymbol{\pi}(\mathbf{Z})} = \boldsymbol{\pi}(\mathbf{Y}) q_{YX}$$

for $\mathbf{X} \sim_j \mathbf{Y}$. If $\mathbf{X} \not\sim_j \mathbf{Y}$, then $q_{XY} = 0 = q_{YX}$, and $\boldsymbol{\pi}(\mathbf{X}) q_{XY} = \boldsymbol{\pi}(\mathbf{Y}) q_{YX}$ trivially. Thus the Markov chain is reversible with respect to the distribution $\boldsymbol{\pi}(\mathbf{X})$, and so $\boldsymbol{\pi}(\mathbf{X})$ is an invariant distribution of this Markov chain, by Lemma 2.16. □

The use of Gibbs sampling is that it enables us to draw samples from a distribution for which sampling would otherwise be infeasible. That is, if the chain we construct is irreducible and aperiodic, then $\boldsymbol{\pi}$ is in fact the unique invariant distribution, and as the process iterates over time, our drawn samples approximate this distribution.

In practice, the indices are picked incrementally instead of at random, and samples are kept only after some burn-in period. This means that given an initial random vector X, for each index i (starting at 1, going to V) we resample x_i. One full cycle is considered an iteration, and the random vector at the end of the cycle is our sample. We "burn in" by ignoring the first N samples (where N is, say, 1000).

The burn-in period removes the effects of the initial random vector and lets the state distribution converge to the unique invariant distribution (which is guaranteed to happen if the chain is irreducible and aperiodic, see Theorem 2.14). After continuing this process, we may stop once we have the desired number of samples.

3.2 Expectation-Maximization

In 1977, Arthur Dempster, Nan Laird, and Donald Rubin published a much lauded paper [6] where they developed the statistical method known as *Expectation-Maximization*. The EM algorithm is a method used to compute the maximum likelihood estimates of parameters in a statistical model in which there are latent (hidden) variables.

In general, given a statistical model with known data O and unobserved data X, unknown parameters λ, and a likelihood function $L(\lambda; X, O) = \mathbb{P}(O, X|\lambda)$ the object is to find λ^* which maximizes $L(\lambda; O)$. Note that we could marginalize out the unobserved data by noting that

$$L(\lambda; O) = \mathbb{P}(O|\lambda) = \sum_X \mathbb{P}(O, X|\lambda)$$

and seek to maximize this, though this computation is often untractable. Instead, we follow an iterative procedure as follows:

(i) Given λ_t, compute $Q(\lambda|\lambda_t) = E_{X|O,\lambda_t}[\ln \mathbb{P}(O, X|\lambda)] = \sum_X \mathbb{P}(X|O, \lambda_t) \ln \mathbb{P}(O, X|\lambda)$.

(ii) Update $\lambda_{t+1} = \underset{\lambda}{\operatorname{argmax}} = Q(\lambda|\lambda_t)$.

The reader should be asking himself the following question: is $\mathbb{P}(O|\lambda_{t+1}) \geq \mathbb{P}(O|\lambda_t)$ necessarily? The answer comes as a result of our development of convex analysis and Jensen's inequality.

First, note that since ln is a strictly increasing function, then the parameter value λ^* which maximizes $l(\lambda) = \ln \mathbb{P}(O|\lambda)$ must also maximize $\mathbb{P}(O|\lambda)$, and so we deal here with the log-likelihood function $l(\lambda)$.

Lemma 3.2. *For any set of parameters λ and current estimate λ_t,*

$$l(\lambda) \geq l(\lambda_t) + \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(O, X|\lambda_t)}.$$

Proof. Note that

$$\begin{aligned}
l(\lambda) - l(\lambda_t) &= \ln \mathbb{P}(O|\lambda) - \ln \mathbb{P}(O|\lambda_t) \\
&= \ln \left(\sum_X \mathbb{P}(O, X|\lambda) \right) - \ln \mathbb{P}(O|\lambda_t) \\
&= \ln \left(\sum_X \mathbb{P}(X|O, \lambda_t) \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(X|O, \lambda_t)} \right) - \ln \mathbb{P}(O|\lambda_t) \\
&\geq \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(X|O, \lambda_t)} - \ln \mathbb{P}(O|\lambda_t) \\
&= \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(X|O, \lambda_t)\mathbb{P}(O|\lambda_t)} \\
&= \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(O, X|\lambda_t)}
\end{aligned}$$

Thus

$$l(\lambda) \geq l(\lambda_t) + \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(O, X|\lambda_t)}.$$

□

Now we can answer our question concerning EM:

Theorem 3.3. *If λ_t and λ_{t+1} are parameter estimates for consecutive iterations of the EM algorithm, then $\mathbb{P}(O|\lambda_{t+1}) \geq \mathbb{P}(O|\lambda_t)$.*

Proof. Note that

$$\begin{aligned}
\lambda_{t+1} &= \operatorname*{argmax}_\lambda \sum_X \mathbb{P}(X|O, \lambda_t) \ln \mathbb{P}(O, X|\lambda) \\
&= \operatorname*{argmax}_\lambda \sum_X \mathbb{P}(X|O, \lambda_t) \ln \mathbb{P}(O, X|\lambda) - \sum_X \mathbb{P}(X|O, \lambda_t) \ln \mathbb{P}(O, X|\lambda_t) \\
&= \operatorname*{argmax}_\lambda \sum_X \mathbb{P}(X|O, \lambda_t) \ln \frac{\mathbb{P}(O, X|\lambda)}{\mathbb{P}(O, X|\lambda_t)}.
\end{aligned}$$

By Theorem 3.2 and this computation we have

$$\begin{aligned}
l(\lambda_{t+1}) &\geq l(\lambda_t) + \sum_X \mathbb{P}(X|O,\lambda_t) \ln \frac{\mathbb{P}(O,X|\lambda_{t+1})}{\mathbb{P}(O,X|\lambda_t)} \\
&\geq l(\lambda_t) + \sum_X \mathbb{P}(X|O,\lambda_t) \ln \frac{\mathbb{P}(O,X|\lambda_t)}{\mathbb{P}(O,X|\lambda_t)} \\
&= l(\lambda_t) + \sum_X \mathbb{P}(X|O,\lambda_t) \ln 1 \\
&= l(\lambda_t).
\end{aligned}$$

The claim is now immediate. □

Given fixed data O, Expectation-Maximization attempts to find the set of parameters to maximize the likelihood of this data, given the assumption that the data was generated from a specific class of models, in the existence of some kind of latent variable. A common use of EM is in training a Gaussian mixture model where only the Gaussian draws are given (the data), but not the components from which each data point is drawn (the latent variables). We will use this procedure to train *Bursting Hidden Markov Models* from data, where certain variables are missing.

It is important to note that while the iterative process is monotonically increasing, this does not guarantee that the method finds a global maximum. In fact, because the monotonicity is not strict, it is even possible that the process converges to a local minimum! A simple approach to dealing with this problem is by using many random restarts. That is, repeat the procedure with varying initial parameter estimates, and settle on the maximizer of all the final estimates. This increases the likelihood that the final trained model is a local maximizer, and hopefully the global.

3.3 Classification Trees

Classification trees are a class of decision trees, developed by Breiman et al in 1984 [7]. They are used in a wide variety of settings where labeled training data is available, and the desired outcome is a model which is able to accurately assign labels to unlabeled data. We assume that each sample d has P attributes, which can be real-valued or categorical, and each sample belongs to some class k, where there are K classes. A toy example of a graphical depiction of a trained classification tree is given in Figure 3.1.

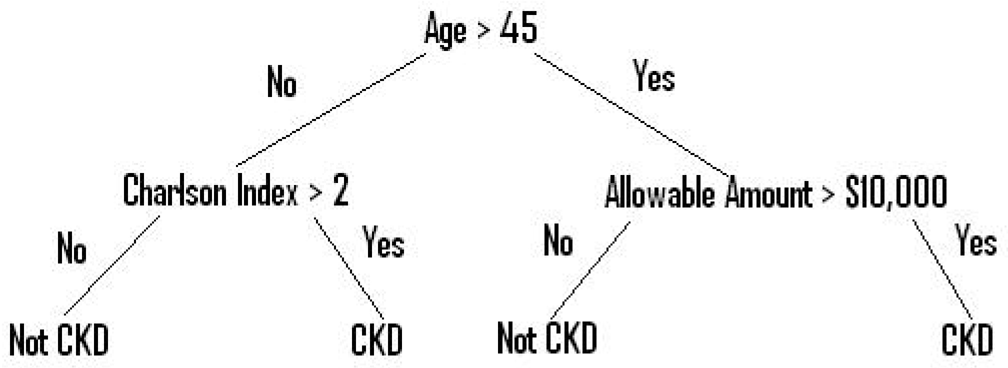

Figure 3.1: Toy decision tree.

Given an unlabeled sample with three variables (age, charlson index, and allowable amount), it is "pushed" down the tree by starting at the *root* node (the top) and performing the following process at each node until arriving at a *leaf* node (a node with no children).

(i) If the expression at the current node is true, then go to the right child node.

(ii) If the expression at the current node is false, then go to the left child node.

Each leaf node has a classification assigned to it, and this unlabeled sample is then labeled with that classification. For example, the unlabeled samples in Table 3.1 will be pushed down the decision tree and receive final labels *Not CKD* and *CKD*, respectively.

The reader may be wondering how a classification tree is trained. We start with a labeled data set D and choose the best attribute p and value x by which to split the data.

Sample #	Age	Charlson	Allowable Amount
1	50	3	\$9,652
2	44	6	\$1,497

Table 3.1: Two unlabeled samples for the toy decision tree.

We have now partitioned D into two sets, which we may then split as well. We continue in this manner until some stopping criteria is met. To formalize this, we need several definitions.

Definition 3.4. Let D be a data set with K different classes. Let N_k be the number of samples labeled class k for each $1 \leq k \leq K$, and let $f_k = \frac{N_k}{N}$ where N is the total number of samples in D. We define the *Gini impurity* to be

$$G(D) = 1 - \sum_{k=1}^{K} f_k^2.$$

Let's examine what this is. Note that each f_k is the fraction of samples in D from class k. But then we have

$$\begin{aligned} G(D) &= 1 - \sum_{k=1}^{K} f_k^2 \\ &= \sum_{k=1}^{K} f_k - f_k^2 \\ &= \sum_{k=1}^{K} f_k(1 - f_k) \end{aligned}$$

But since f_k can also be considered the probability of randomly selecting a sample with class k from D and $(1 - f_k)$ can be considered the probability of incorrectly labeling a sample with class k, this means that $G(D)$ is a measure of how often one would incorrectly label a randomly chosen sample from D. Other measures of impurity exist, but in our implementation we use the Gini impurity.

We now define what it means to be a *split* of a data set D.

Definition 3.5. We define the *split* $s_D(p, x)$ of the data set D on attribute p using value x, to be a partition D_1, D_2 such that

(i) $d_p \leq x$ for all $d \in D_1$ and $d_p > x$ for all $d \in D_2$, where d_p is the value of attribute p in d, assuming real values; or

(ii) $d_p \neq x$ for all $d \in D_1$ and $d_p = x$ for all $d \in D_2$, where d_p is the value of attribute p in d, assuming categorical values.

Note that $D_1 \cap D_2 = \emptyset$ and $D = D_1 \cup D_2$, so $s_D(p, x)$ is in fact a partition of D.

Given a set of labeled samples D, we wish to find a split that maximizes *information gain.*

Definition 3.6. Let $s_D(p, x) = D_1, D_2$ be a split. We define the *information gain* of this split to be

$$I(s_D(p, x)) = G(D) - \sum_{i=1}^{2} \frac{|D_i|}{|D|} \cdot G(D_i).$$

Thus the information gain is the difference between the Gini impurity of the parent set and the weighted sum of the Gini impurities of the child sets.

We consider each node to be associated with a data set D. The root node is associated with all available training data. To train a classification tree, given a node we find

$$s_D^\star = \operatorname*{argmax}_{p,x} I(s_D(p, x))$$

and create two child nodes. The left child node is associated with D_1 and the right child node with D_2. We continue splitting nodes until each leaf node is pure, i.e. consists of only a single class. At this point each leaf node is classified with the class label it contains.

In practice, classification trees are *pruned*, meaning that they are not grown out to complete purity. In this case, a leaf node contains a mixture of classes, and is labeled with the majority class. In our implementation of the random forest, our trees are not pruned, so we offer no further discussion of pruning here.

3.4 Random Forests

Random forests are aptly named; they are collections of classification trees, an extension made by Breiman of his own work [8]. The difference is that they introduce an amount of randomness in the inference procedure, whereas classification trees are deterministic based on the training data. To be more precise, let D be the set of labeled samples available for training, K the number of distinct classes, P the number of attributes of each data point, and $p << P$, a user-specified natural number significantly smaller than P. Then a random forest is a collection of T trees, where splits at each node are performed in the following manner:

(i) Randomly select p attributes.

(ii) Compute the optimal split of the available data, restricting splits to the p selected attributes.

Thus for each split, only a random sampling of attributes can be considered. Each tree is fully grown, i.e. not pruned. To predict the label for a new sample d, each tree in the random forest assigns d a label as discussed above. For each possible label k, let $\widehat{N_k}$ be the number of trees assigning d label k. We can view the prediction from the random forest in two ways:

(i) Label d with $\operatorname{argmax}_k N_k$; or

(ii) Label d with a probability distribution $\mathbf{p}$ over the classes, where $\mathbf{p}_k = \frac{|N_k|}{|T|}$.

In our Chronic Kidney Disease classifier, we follow the second option, enabling us to use ROC analysis to measure the efficiency of the classifier.

3.5 Receiver Operating Characteristic

One persistent problem in supervised learning (classification) is how to best deal with rare data. In a classification problem, rare data denotes the presence of a class for which there is very little data, whereas a plethora of data exists for other classes (called the majority classes). This is often a challenge when training classifiers on medical conditions, since the condition of interest may be relatively rare in the overall population. In general, left untreated, a classifier will naturally give weight to the majority class(es), making it a poor predictor of the rare class of interest.

A number of ways of dealing with rarity have been proposed and implemented over the years. These include oversampling the rare class, undersampling the majority class(es), and cost-sensitive learning. In a binary classification problem, a good way to deal with rarity is using receiver operating characteristic (ROC) analysis. For the following definitions, we assume that in a data set D, each sample is labeled with one of two classes: positive or negative. Throughout we assume that the positive class is rare, and the negative class is majority.

Definition 3.7. Given a classifier, a sample is a *true positive* if it is positive and is correctly labeled as such. A sample is a *false positive* if it is negative but is labeled as positive. A sample is a *true negative* if it is negative and is correctly labeled as such, and is a *false negative* if it is positive and incorrectly labeled as negative. Given a sample set and classifier, let N_{tp} denote the number of samples that are true positive, N_{fp} the number of samples that are false positive, N_{tn} the number of samples that are true negative, and N_{fn} the number of samples that are false negative.

Definition 3.8. Given a classifier and sample set D, the *sensitivity* is the fraction of true positives over all positives, and the *specificity* is the fraction of true negatives over all negatives. More specifically,

$$\text{Sens}_D = \frac{N_{tp}}{N_{tp} + N_{fn}}$$
$$\text{Spec}_D = \frac{N_{tn}}{N_{tn} + N_{fp}}$$

Essentially, Sens_D is the probability of a positive being labeled correctly, and Spec_D is the probability of a negative being labeled correctly. For example, suppose we have a sample set with 100 positives and 165,000 negatives, and a classifier with a sensitivity $\text{Sens}_D = 0.09$ and a specificity $\text{Spec}_D = 0.93$. Then we would correctly classify 9 people as positive and 153,450 as negative, while incorrectly classfying 91 positives as negatives and 11,550 negatives as positives, i.e. we would have 91 false negatives and 11,550 false positives.

Often with a binary classifier, for each new sample, it is possible to get a probability distribution over the two class labels instead of a single hard label assignment. If the positive class is rare, then often these classifiers will naturally tend to predict a greater probability of a sample being the majority class, even if this is incorrect. This yields a high specificity, but a very low sensitivity. It may be that we are willing to have more overall misclassifications, if in return we are able to correctly identify the rare classes, i.e. we may be willing to sacrifice specificity in return for sensitivity. To do this, we define a specificity-sensitivity pair.

Definition 3.9. The *specificity-sensitivity pair* for a classifier at $\lambda \in [0, 1]$ for a sample set D is

$$\text{Spec-Sens}_D(\lambda) = (\text{Spec}_D, \text{Sens}_D)\,,$$

where each sample $d \in D$ is classified as positive if $\mathbb{P}(d \text{ is positive}) \geq \lambda$ and negative otherwise.

ROC analysis allows us to see how the specificity and sensitivity are related as we vary λ between 0 and 1. This is usually depicted by a *ROC curve*, in which one minus the specificity is plotted on the x-axis against the sensitivity on the y-axis, as in Figure 3.2.

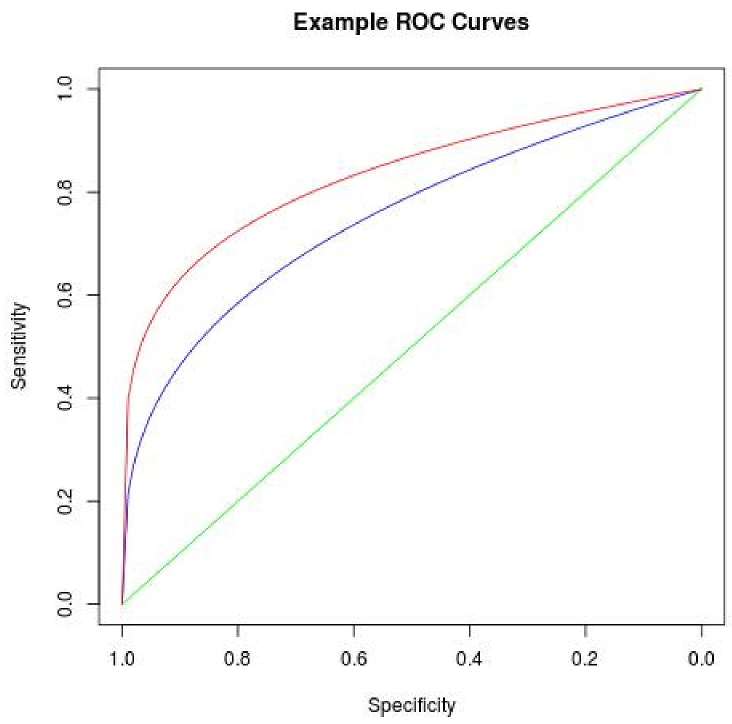

Figure 3.2: ROC curve examples with base line.

Of these classifiers, clearly the one with overall higher specificity-sensitivity pairs is a superior classifier, as it results in fewer misclassifications. This provides a measure of the performance of a binary classifier: the area under the curve.

Definition 3.10. Let f be a ROC curve, which we may consider to be a function $f : [0, 1] \to [0, 1]$ where $f(0) = 0$ and $f(1) = 1$. Define the *ROC score* of f to be

$$RS(f) = \int_0^1 f(x)dx$$

We will use this performance measure to compare the quality of different binary classifiers in Chapter 6.

Developed in 2003 by Blei et al [9], Latent Dirichlet Allocation (LDA) replaced other topic analysis methods such as Latent Semantic Indexing (LSI) [10], probabilistic Latent Semantic Indexing (pLSI) [11], and others. These tools had been developed to extract topical information from large collections of texts. The idea was that they could categorize each document in a collection as being about topic X or topic Y, thus enhancing information retrieval methods. Given a specific document d out of a corpus (collection of documents), one could use these tools to identify which documents are most similar to document d according to some well defined metric.

LDA took this a step further, being significantly different from previous tools in that it provided a generative model for corpus creation. It also allowed us to characterize documents as being a distribution over topics, i.e. document i might be 40% topic X and 60% topic Y. This has obvious advantages. While we will ultimately use LDA in dealing with patients' health care claims and diagnosis codes, we present it in its original context of text analysis.

4.1 Formalization and Background

To formalize terminology, we refer to a *term* as a specific sequence of lowercase alphanumeric characters with some defined meaning. A *document* is a finite list of *words*, where each word is a specific instance of a term, allowing repetition. A *corpus* is a finite list of documents. A *term list* for a corpus is a list of all terms that occur in at least one document in the corpus.

Throughout, we make use of two important distributions: categorical and symmetric Dirichlet. While the categorical distribution is quite common, the symmetric Dirichlet distribution is not, so we provide an introduction.

4.1.1 Dirichlet Distribution. The symmetric Dirichlet distribution takes as a parameter a scalar $\alpha \in \mathbb{R}, \alpha > 0$. In general, a K-dimensional Dirichlet distribution has a vector parameter $\boldsymbol{\alpha} \in \mathbb{R}^K$. Given K rival events, where for each i the i^{th} event has occured $\alpha_i - 1$

times, the probability density function of Dir($\boldsymbol{\alpha}$) returns the belief that the probability distribution of the K events is $\mathbf{x}$. This can be explicitly written as below

$$f(\mathbf{x};\alpha) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

and if the distribution is the symmetric Dirichlet distribution, this simplifies to

$$f(\mathbf{x};\alpha) = \frac{\Gamma(K\alpha)}{K\Gamma(\alpha)}(\prod_{i=1}^{K} x_i)^{\alpha-1}.$$

Note specifically that if $\alpha = 1$, then $(\prod_{i=1}^{K} x_i)^{\alpha-1} = (\prod_{i=1}^{K} x_i)^0 = 1$, so $f(\mathbf{x}_1;1) = f(\mathbf{x}_2;1)$ for any distributions $\mathbf{x}_1, \mathbf{x}_2$. Since for each $\mathbf{x}$ we have that $x_i > 0$ for all i and $\sum_{i=1}^{K} x_i = 1$, we note that $\mathbf{x}$ lives in the standard open $K-1$-simplex.

Some examples of the probability density functions for a 3-dimensional Dirichlet distribution are given in Figure 4.1.

Note how for symmetric parameters with $\alpha < 1$, the distribution gets sparse, in that the categorical distributions drawn from such a Dirichlet distribution are weighted heavily in only one dimension.

4.2 Text Generation via Latent Dirichlet Allocation

Latent Dirichlet Allocation assumes that each corpus of M documents over K topics with a vocabulary size V, is generated by a model having several parameters: $\eta \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^K, \boldsymbol{\beta} \in \mathbb{R}^V$. Here, η is a parameter for a Poisson distribution, representing the tendency of certain lengths for each document, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are Dirichlet parameters, used to generate categorical distributions over the K topics and the V terms, respectively. LDA further assumes that each corpus of M documents is generated via Algorithm 4.1:

This generation process is often depicted graphically with plate notation as shown in Figure 4.2.

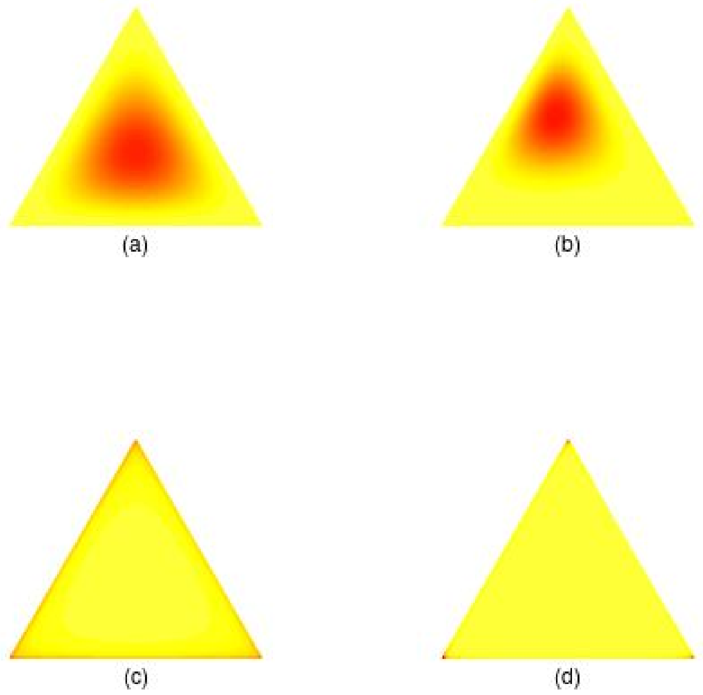

Figure 4.1: Dirichlet distributions with various parameter vectors α, represented by a heat map. Clockwise from top left: $\boldsymbol{\alpha} = (3,3,3), (3,6,4), (.95,.95,.95), (.3,.3,.3)$.

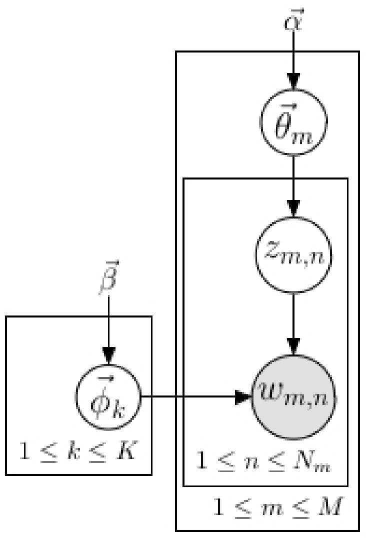

Figure 4.2: Latent Dirichlet Allocation graphical plate notation.

Algorithm 4.1 Given Dirichlet parameters $\boldsymbol{\alpha} \in \mathbb{R}^K$, $\boldsymbol{\beta} \in \mathbb{R}^V$, and a Poisson parameter $\eta \in \mathbb{R}$, generate a corpus of M documents.

```
for $k = 1 \to K$ do
    Sample $\boldsymbol{\phi}_k \sim \text{Dir}(\boldsymbol{\beta})$
end for
for $m = 1 \to M$ do
    Sample $N_m \sim \text{Poiss}(\eta)$
    Sample $\boldsymbol{\theta}_m \sim \text{Dir}(\boldsymbol{\alpha})$
    for $n = 1 \to N_m$ do
        Sample $\boldsymbol{z}_{m,n} \sim \text{Cat}(\boldsymbol{\theta}_m)$
        Sample $\boldsymbol{w}_{m,n} \sim \text{Cat}(\boldsymbol{\phi}_{\boldsymbol{z}_{m,n}})$
    end for
end for
```

We refer to $\boldsymbol{\theta}_m$ as the *topic distribution* for document m, and $\boldsymbol{\phi}_k$ as the *term distribution* for topic k. That is, $\boldsymbol{\theta}_m$ is a categorical distribution of length K, such that

$$\theta_{m,k} = \mathbb{P}(z = k|d = m),$$

i.e., the probability of seeing topic k, given that we are considering document m. Similarly, $\boldsymbol{\phi}_k$ is a categorical distribution of length V, such that

$$\phi_{k,v} = \mathbb{P}(w = v|z = k),$$

i.e., the probability of seeing term v, given that we are considering topic k. Throughout we write $\boldsymbol{\Theta}$ as the matrix with m^{th} row $\boldsymbol{\theta}_m$ and $\boldsymbol{\Phi}$ as the matrix with k^{th} row $\boldsymbol{\phi}_k$. We also write $\boldsymbol{w}$ to denote the set of all words in the corpus, $\boldsymbol{w}_m$ being the set of words in document m, and $w_{m,n}$ being the n^{th} word in document m, and $\boldsymbol{z}$ to denote the set of all topic assignments in the corpus, $\boldsymbol{z}_m$ being the set of topic assignments in document m, and $z_{m,n}$ being the topic assignment for word $w_{m,n}$.

The power of LDA lays in our ability to discover the *latent* variables $\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{z}$ given a corpus of documents. While the generation of documents according to the assumptions of LDA is fairly straightforward, their inference is generally intractable. Thus we rely on a Gibbs sampling algorithm to infer the hidden variables given a corpus. Other methods (such as variational expectation-maximization [9]) exist, but the simplicity of Gibbs sampling makes it the ideal choice for this introductory work.

4.3 Latent Dirichlet Allocation Inference

The derivations in this section closely mirror the work in [12] and [13], and we include it here for self-containment. We wish to sample from $\mathbb{P}(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, but this is in general intractable. Thus we use a Gibbs sampler, and for each word b in each document a we must sample from $\mathbb{P}(z_{a,b}|\boldsymbol{z}_{\neg a,b}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ where $\boldsymbol{z}_{\neg a,b}$ denotes all the topic assignments to all words other than the b^{th} word in the a^{th} document. Before we present the derivation of this conditional distribution, we must make a few additional definitions, hereafter referred to as the *count matrices*:

$$n_{(k,m,v)} = \text{ The number of appearances of the term } v \text{ in document } m \text{ assigned to topic } k. \tag{4.1}$$

$$n_{(k,m,\cdot)} = \sum_{v=1}^{V} n_{(k,m,v)} = \text{ The number of words in document } m \text{ assigned to topic } k. \tag{4.2}$$

$$n_{(k,\cdot,v)} = \sum_{m=1}^{M} n_{(k,m,v)} = \text{ The number of times term } v \text{ is assigned to topic } k. \tag{4.3}$$

$$n^{\neg}_{(k,a,\cdot)} = n_{(k,a,\cdot)} - \mathbb{1}_{[z_{a,b}=k]} \tag{4.4}$$

$$n^{\neg}_{(k,\cdot,v)} = n_{(k,\cdot,v)} - \mathbb{1}_{[w_{a,b}=v]} \tag{4.5}$$

4.3.1 Some Computations. We now make a few computations to make the future work easier. Note that by the law of total probability and integrating out $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ we have

$$\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int\int \mathbb{P}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) d\boldsymbol{\Theta} d\boldsymbol{\Phi}. \tag{4.6}$$

Also by conditional probabilities,

$$\mathbb{P}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\Theta}, \boldsymbol{\Phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbb{P}(\boldsymbol{\Phi}|\boldsymbol{\beta})\mathbb{P}(\boldsymbol{\Theta}|\boldsymbol{\alpha})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\Theta})\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\Phi}). \tag{4.7}$$

Noting that the documents are independent from each other and the topics are independent of each other, we have

$$\mathbb{P}(\boldsymbol{z}|\boldsymbol{\Theta})\mathbb{P}(\boldsymbol{\Theta}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} \mathbb{P}(\boldsymbol{z}_m|\boldsymbol{\theta}_m)\mathbb{P}(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) \tag{4.8}$$

and

$$\mathbb{P}(\boldsymbol{\Phi}|\boldsymbol{\beta})\mathbb{P}(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\Phi}) = \prod_{k=1}^{K} \mathbb{P}(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathbb{P}(w_{m,n}|\boldsymbol{\phi}_{z_{m,n}}) \tag{4.9}$$

Since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are Dirichlet distributions, we have

$$\mathbb{P}(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k - 1} \tag{4.10}$$

and

$$\mathbb{P}(\boldsymbol{\phi}_k|\boldsymbol{\beta}) = \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \tag{4.11}$$

From the definition of multinomial distributions, we also have

$$\mathbb{P}(\boldsymbol{z}_m|\boldsymbol{\theta}_m) = \prod_{n=1}^{N_m} \theta_{m,z_{m,n}} \tag{4.12}$$

and

$$\mathbb{P}(w_{m,n}|\boldsymbol{\phi}_{z_{m,n}}) = \phi_{z_{m,n},w_{m,n}} \tag{4.13}$$

From the definitions of $n_{(k,m,\cdot)}$ and $n_{(k,\cdot,v)}$ note that

$$\prod_{n=1}^{N_m} \theta_{m,z_{m,n}} = \prod_{k=1}^{K} \theta_{m,k}^{n_{(k,m,\cdot)}} \tag{4.14}$$

and

$$\prod_{m=1}^{M}\prod_{n=1}^{N_m}\phi_{z_{m,n},w_{m,n}} = \prod_{v=1}^{V}\phi_{k,v}^{n_{(k,\cdot,v)}} \tag{4.15}$$

With these definitions and computations, we may proceed with the derivation of the conditional distribution $\mathbb{P}(z_{a,b}|\boldsymbol{z}_{\neg a,b},\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta})$.

4.3.2 Derivation of the Conditional Distribution. By the definition of conditional probability,

$$\mathbb{P}(z_{a,b}|\boldsymbol{z}_{\neg a,b},\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{\mathbb{P}(z_{a,b},\boldsymbol{z}_{\neg a,b},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}{\mathbb{P}(\boldsymbol{z}_{\neg a,b},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta})}$$

Since the denominator is independent of $z_{a,b}$,

$$\propto \mathbb{P}(z_{a,b},\boldsymbol{z}_{\neg a,b},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta})$$

Now since $\boldsymbol{z} = \{z_{a,b}\}\cup\{\boldsymbol{z}_{\neq,a,b}\}$, this becomes

$$= \mathbb{P}(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\beta})$$

By 4.6 and 4.7,

$$\begin{aligned}&= \int\int \mathbb{P}(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{\alpha},\boldsymbol{\beta})d\boldsymbol{\Theta}d\boldsymbol{\Phi}\\ &= \int\int \mathbb{P}(\boldsymbol{\Phi}|\boldsymbol{\beta})\mathbb{P}(\boldsymbol{\Theta}|\boldsymbol{\alpha})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\Theta})\mathbb{P}(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\Phi})d\boldsymbol{\Theta}d\boldsymbol{\Phi}\end{aligned}$$

Separating the integrals, this becomes

$$= \int \mathbb{P}(\boldsymbol{z}|\boldsymbol{\Theta})\mathbb{P}(\boldsymbol{\Theta}|\boldsymbol{\alpha})d\boldsymbol{\Theta} \times \int \mathbb{P}(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\Phi})\mathbb{P}(\boldsymbol{\Phi}|\boldsymbol{\beta})d\boldsymbol{\Phi}$$

By 4.8 and 4.9 we have

$$= \int \prod_{m=1}^{M} \mathbb{P}(\boldsymbol{z}_m|\boldsymbol{\theta}_m)\mathbb{P}(\boldsymbol{\theta}_m|\boldsymbol{\alpha})d\boldsymbol{\Theta} \times \int \prod_{k=1}^{K} \mathbb{P}(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{m=1}^{M}\prod_{n=1}^{N_m} \mathbb{P}(w_{m,n}|\boldsymbol{\phi}_{z_{m,n}})d\boldsymbol{\Phi}$$

Again, by the independence of documents from each other and topics from each other, this becomes

$$= \prod_{m=1}^{M} \int \mathbb{P}(\boldsymbol{z}_m|\boldsymbol{\theta}_m)\mathbb{P}(\boldsymbol{\theta}_m|\boldsymbol{\alpha})d\boldsymbol{\theta}_m \times \prod_{k=1}^{K} \int \mathbb{P}(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \prod_{m=1}^{M}\prod_{n=1}^{N_m} \mathbb{P}(w_{m,n}|\boldsymbol{\phi}_{z_{m,n}})d\boldsymbol{\phi}_k$$

Then by 4.10 through 4.13, we have

$$= \prod_{m=1}^{M} \int \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k-1} \prod_{n=1}^{N_m} \theta_{m,z_{m,n}} d\boldsymbol{\theta}_m \times \prod_{k=1}^{K} \int \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v-1} \prod_{m=1}^{M}\prod_{n=1}^{N_m} \phi_{z_{m,n},w_{m,n}} d\boldsymbol{\phi}_k$$

and by 4.14 and 4.15 and pulling constants outside the integrals, this becomes

$$= \prod_{m=1}^{M} \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \int \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k+n_{(k,m,\cdot)}-1} d\boldsymbol{\theta}_m \times \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \int \prod_{v=1}^{V} \phi_{k,v}^{\beta_v+n_{(k,\cdot,v)}-1} d\boldsymbol{\phi}_k$$

Cleverly multiplying each integral by "1", we find

$$\begin{aligned} = \prod_{m=1}^{M} \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \frac{\prod_{k=1}^{K}\Gamma(n_{(k,m,\cdot)}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}n_{(k,m,\cdot)}+\alpha_k)} \int \frac{\Gamma(\sum_{k=1}^{K}n_{(k,m,\cdot)}+\alpha_k)}{\prod_{k=1}^{K}\Gamma(n_{(k,m,\cdot)}+\alpha_k)} \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k+n_{(k,m,\cdot)}-1} d\boldsymbol{\theta}_m \\ \times \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)} \frac{\prod_{v=1}^{V}\Gamma(n_{(k,\cdot,v)}+\beta_v)}{\Gamma(\sum_{v=1}^{V}n_{(k,\cdot,v)}+\beta_v)} \int \frac{\Gamma(\sum_{v=1}^{V}n_{(k,\cdot,v)}+\beta_v)}{\prod_{v=1}^{V}\Gamma(n_{(k,\cdot,v)}+\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v+n_{(k,\cdot,v)}-1} d\boldsymbol{\phi}_k \end{aligned}$$

But the integrands are now the pdfs of the Dirichlet distributions with parameters

$$\boldsymbol{\alpha}' = (\alpha_1 + n_{(1,m,\cdot)}, \cdots, \alpha_K + n_{(K,m,\cdot)})$$

and

$$\boldsymbol{\beta}' = (\beta_1 + n_{(k,\cdot,1)}, \cdots, \beta_V + n_{(k,\cdot,V)})$$

respectively over their entire support, so they evaluate to 1, leaving

$$= \prod_{m=1}^{M} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(n_{(k,m,\cdot)} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{(k,m,\cdot)} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \frac{\prod_{v=1}^{V} \Gamma(n_{(k,\cdot,v)} + \beta_v)}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)}$$

Dropping the constants dependent only on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and hence conditionally independent of $z_{a,b}$, this now becomes

$$\propto \prod_{m=1}^{M} \frac{\prod_{k=1}^{K} \Gamma(n_{(k,m,\cdot)} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{(k,m,\cdot)} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{(k,\cdot,v)} + \beta_v)}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)}$$

Separate the terms involving a and b from everything else, so we have

$$= \left[\prod_{m \neq a} \frac{\prod_{k=1}^{K} \Gamma(n_{(k,m,\cdot)} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{(k,m,\cdot)} + \alpha_k)}\right] \times \frac{\prod_{k=1}^{K} \Gamma(n_{(k,a,\cdot)} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{(k,a,\cdot)} + \alpha_k)}$$
$$\times \prod_{k=1}^{K} \frac{\left[\prod_{v \neq w_{a,b}} \Gamma(n_{(k,\cdot,v)} + \beta_v)\right] \times \Gamma(n_{(k,\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)}$$

and dropping the constants independent of a and b, it becomes

$$\propto \frac{\prod_{k=1}^{K} \Gamma(n_{(k,a,\cdot)} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} n_{(k,a,\cdot)} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\Gamma(n_{(k,\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)}$$

Now considering definitions 4.4 and 4.5 and separating topic $z_{a,b}$ from the other topics, we have

$$\propto \frac{\prod_{k \neq z_{a,b}} \Gamma(n_{(k,a,\cdot)} + \alpha_k) \times \Gamma(n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}} + 1)}{\Gamma(1 + \sum_{k=1}^{K} n^{\neg}_{(k,a,\cdot)} + \alpha_k)}$$
$$\times \prod_{k \neq z_{a,b}} \frac{\Gamma(n_{(k,\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)} \times \frac{\Gamma(n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}} + 1)}{\Gamma(1 + \sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)} + \beta_v)}$$

Noting that a property of the gamma function is $\Gamma(n+1) = n\Gamma(n)$, we can separate this further as

$$= \frac{\prod_{k \neq z_{a,b}} \Gamma(n_{(k,a,\cdot)} + \alpha_k) \times \Gamma(n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}}) \times (n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}})}{\Gamma(1 + \sum_{k=1}^{K} n^{\neg}_{(k,a,\cdot)} + \alpha_k)}$$
$$\times \prod_{k \neq z_{a,b}} \frac{\Gamma(n_{(k,\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n_{(k,\cdot,v)} + \beta_v)} \times \frac{\Gamma(n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}}) \times (n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)} + \beta_v) \times (\sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)} + \beta_v)}$$

Reconsolidating and recognizing that $n_{(k,\cdot,v)} = n^{\neg}_{(k,\cdot,v)}$ for $k \neq z_{a,b}$, we have

$$= \frac{\prod_{k=1}^{K} \Gamma(n^{\neg}_{(k,a,\cdot)} + \alpha_k) \times (n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}})}{\Gamma(1 + \sum_{k=1}^{K} n^{\neg}_{(k,a,\cdot)} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\Gamma(n^{\neg}_{(k,\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\Gamma(\sum_{v=1}^{V} n^{\neg}_{(k,\cdot,v)} + \beta_v)} \times \frac{n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}}}{\sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)} + \beta_v}$$

Dropping constants (everything independent of $z_{a,b}$) this becomes

$$\propto \frac{(n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}}) \times (n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}})}{\sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)} + \beta_v}$$

and rewriting $\sum_{v=1}^{V} n^{\neg}_{(z_{a,b},\cdot,v)}$ as $n^{\neg}_{(z_{a,b},\cdot,\cdot)}$ this is

$$\propto \frac{(n^{\neg}_{(z_{a,b},a,\cdot)} + \alpha_{z_{a,b}}) \times (n^{\neg}_{(z_{a,b},\cdot,w_{a,b})} + \beta_{w_{a,b}})}{n^{\neg}_{(z_{a,b}\cdot,\cdot)} + \sum_{v=1}^{V} \beta_v}$$

Note here that the first multiplicand of the numerator is the number of words in document a which have been assigned to the topic $z_{a,b}$ (excluding the b^{th} word itself) plus the prior for the topic, $\alpha_{z_{a,b}}$. The second multiplicand in the numerator is the number of times the term $w_{a,b}$ is assigned to the topic $z_{a,b}$ in the whole corpus (excluding the b^{th} word of the a^{th} document) plus the prior for the term $w_{a,b}$. The denominator is the total number of words in the corpus assigned to the topic $z_{a,b}$ (excluding the b^{th} word of the a^{th} document) plus the sum of the term priors. Since we have shown that this distribution is only proportional to this final value, we must of course normalize this before sampling, making the final conditional distribution

$$\mathbb{P}(z_{a,b}|\boldsymbol{z}_{\neg,a,b},\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{\frac{(n^{\neg}_{(z_{a,b},a,\cdot)}+\alpha_{z_{a,b}})\times(n^{\neg}_{(z_{a,b},\cdot,w_{a,b})}+\beta_{w_{a,b}})}{n^{\neg}_{(z_{a,b}\cdot,\cdot)}+\sum_{v=1}^{V}\beta_v}}{\sum_{k=1}^{K}\frac{(n^{\neg}_{(k,a,\cdot)}+\alpha_k)\times(n^{\neg}_{k,\cdot,w_{a,b})}+\beta_{w_{a,b}})}{n^{\neg}_{(k\cdot,\cdot)}+\sum_{v=1}^{V}\beta_v}} \tag{4.16}$$

This final version is easy to sample from, since we can easily keep track of the count matrices throughout the entire Gibbs sampling process for all documents in the corpus. Considering our earlier discussion of Markov chains and Gibbs sampling, we should check that our conditions for convergence to a unique invariant distribution are met.

We must recognize that the state space is the set of all possible $\boldsymbol{z}$, so it is K^N, where we identify K with the set $\{1,2,\cdots,K\}$ and N is the total number of words in the corpus. Thus it is finite. Note also that each conditional distribution has only positive entries; this is because α_k and β_v is positive for each k and v, which is a requirement for Dirichlet parameters. Thus, given any two states $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, there is a positive probability that state $\boldsymbol{z}_1$ will transition to $\boldsymbol{z}_2$ in N iterations. Thus the Markov chain induced by the Gibbs sampler is irreducible.

The chain is also aperiodic. This is clearly true, since given the current state $\boldsymbol{z}_1$ and an index i, there is a positive probability that the the i^{th} coordinate of $\boldsymbol{z}_1$ will remain unchanged, yielding period 1.

Since the Gibbs sampler creates a Markov chain with an invariant distribution, we may use 2.14 to show that the process converges to a unique invariant distribution. Thus in our sampling process, we are indeed approximately sampling from the desired distribution $\mathbb{P}(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta})$.

4.3.3 Re-estimation. Once we have burned-in in the process and have taken a sample $\boldsymbol{z}$, we can examine the counts $n_{(k,m,\cdot)}$ and $n_{(k,\cdot,v)}$ to reestimate $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ as follows:

$$\widehat{\theta_{m,k}} = \frac{n_{(k,m,\cdot)} + \alpha_k}{n_{(\cdot,m,\cdot)} + \sum_{k=1}^{K} \alpha_k} \tag{4.17}$$

$$\widehat{\phi_{k,v}} = \frac{n_{(k,\cdot,v)} + \beta_v}{n_{(k,\cdot,\cdot)} + \sum_{v=1}^{V} \beta_v} \tag{4.18}$$

Sample $\boldsymbol{z}$ a sufficient number of times, aggregate the count matrices, and compute $\widehat{\boldsymbol{\Theta}}$ and $\widehat{\boldsymbol{\Phi}}$ to find a good estimation of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ for the corpus. We summarize this in Algorithm 4.2.

Algorithm 4.2 Given a corpus of M documents, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, estimate $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$

```
Let $n'_{(k,m,\cdot)}, n'_{(k,\cdot,v)}, n'_{(\cdot,m,\cdot)}, n'_{(k,\cdot,\cdot)} = 0$ for all topics $k$, terms $v$, and documents $m$.
for $m = 1 \to M$ do
    for $n = 1 \to N_m$ do
        Sample $z_{m,n} = k \sim \text{Cat}(\frac{1}{K})$
        Increment $n'_{(k,m,\cdot)}, n'_{(k,\cdot,w_{m,n})}, n'_{(\cdot,m,\cdot)}, n'_{(k,\cdot,\cdot)}$
    end for
end for
while Not converged do
    for $m = 1 \to M$ do
        for $n = 1 \to N_m$ do
            Decrement $n'_{(k,m,\cdot)}, n'_{(k,\cdot,w_{m,n})}, n'_{(\cdot,m,\cdot)}, n'_{(k,\cdot,\cdot)}$ where $k = z_{m,n}$
            Sample $z_{m,n} = \overline{k} \sim \mathbb{P}(z_{m,n}|\boldsymbol{z}_{\neg,m,n}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ as given in 4.16
            Increment $n'_{(\overline{k},m,\cdot)}, n'_{(\overline{k},\cdot,w_{m,n})}, n'_{(\cdot,m,\cdot)}, n'_{(\overline{k},\cdot,\cdot)}$
        end for
    end for
    Check convergence
end while
Define $n_{(k,m,\cdot)}, n_{(k,\cdot,v)}, n_{(\cdot,m,\cdot)}, n_{(k,\cdot,\cdot)} = 0$ for all topics $k$, terms $v$, and documents $m$.
for $i = 1 \to L$ do
    Update $n_{(k,m,\cdot)} = n_{(k,m,\cdot)} + n'_{(k,m,\cdot)}$
    Update $n_{(k,\cdot,v)} = n_{(k,\cdot,v)} + n'_{(k,\cdot,v)}$
    Update $n_{(\cdot,m,\cdot)} = n_{(\cdot,m,\cdot)} + n'_{(\cdot,m,\cdot)}$
    Update $n_{(k,\cdot,\cdot)} = n_{(k,\cdot,\cdot)} + n'_{(k,\cdot,\cdot)}$
end for
Estimate $\widehat{\boldsymbol{\Theta}}$ and $\widehat{\boldsymbol{\Phi}}$ according to 4.17 and 4.18 respectively
return $\widehat{\boldsymbol{\Theta}}$ and $\widehat{\boldsymbol{\Phi}}$
```

Thus we may estimate the topic distributions of each document in a corpus and the term distribution for each topic as well, given the words in the corpus and the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. In practice, good results can often still be achieved with using symmetric parameters with small values, say 0.1 for each. Throughout our later analysis, we adopt the approach of only considering symmetric Dirichlet parameters, varying their values and the number of topics to find a good classifier.

Chapter 5. Bursting Hidden Markov Models

We first present the standard Hidden Markov Model (HMM), explain it as a generative process, and give three classical problems associated with HMMs. We then introduce an extension, and provide solutions to the three problems for this generalized model.

5.1 Hidden Markov Models

In a Hidden Markov Model, there is a homogeneous Markov chain $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots\}$ assuming values in a finite state space Q with transition matrix A, as well as another discrete-time process in which a sequence of random variables $\boldsymbol{O} = \{O_1, O_2, \cdots\}$ takes values from the finite observation space $S = \{1, \cdots, M\}$. This process must also be homogeneous, in that

$$\mathbb{P}(O_t = s | \mathbf{x}_t = i) = \mathbb{P}(O_1 = s | \mathbf{x}_1 = i) \text{ for all } t \in \mathbb{N}, i \in Q, s \in S. \tag{5.1}$$

In other words, the probability of any observation is only state independent, and not time dependent. Also, similar to state transitions, these observation emissions are dependent only on the current state. We can graphically represent this process with the following diagram:

$$\begin{tikzcd}
\mathbf{x}_1 \arrow[r] \arrow[d] & \mathbf{x}_2 \arrow[r] \arrow[d] & \cdots \arrow[r] \arrow[d] & \mathbf{x}_{t-1} \arrow[r] \arrow[d] & \mathbf{x}_t \arrow[r] \arrow[d] & \mathbf{x}_{t+1} \arrow[r] \arrow[d] & \cdots \arrow[d] \\
O_1 & O_2 & \cdots & O_{t-1} & O_t & O_{t+1} & \cdots
\end{tikzcd}$$

Since the process is homogeneous and the observation emissions depend only on the current state, this allows us to model the observation emissions by an $N \times M$ row stochastic observation matrix $B = (b_{ij})$, where

$$b_{ij} = \mathbb{P}(O_t = j|\mathbf{x}_t = i). \tag{5.2}$$

Thus b_{ij} is the probability of observing j given that the current state is i. We let B_i denote the i^{th} row of B.

Finally, to complete this model, an initial state distribution is required. This is typically represented by a $N \times 1$ stochastic vector $\boldsymbol{\pi} = (\pi_i)$ where

$$\pi_i = \mathbb{P}(\mathbf{x}_1 = i). \tag{5.3}$$

We now define $\lambda = (A, B, \boldsymbol{\pi})$ to be a Hidden Markov Model. It is "hidden" in that HMMs are used to analyze data when the observation sequence is known but the underlying state sequence is unknown. Their power is in their ability to provide probabilistic information about a state sequence based solely on the model and a given observation sequence, though their strengths transcend even further. Hidden Markov Models are a powerful pattern recognition tool, and have been employed in speech recognition, handwriting recognition, cryptanalysis, and bioinformatics, to name a few.

5.2 HMM Generation

It is often useful to view models by their generative process, so here we offer a simple explanation of the generative process of an HMM. Given a desired number of transitions T, follow the given generative algorithm:

Algorithm 5.1 Given the HMM λ, generate an observation sequence $\mathbf{O}$

```
$\mathbf{x}_1 \sim \mathrm{Cat}(\boldsymbol{\pi})$                ▷ Sample $\mathbf{x}_1$ from the initial state distribution $\boldsymbol{\pi}$
for $t = 1 \to T$ do
    $O_t \sim \mathrm{Cat}(B_{\mathbf{x}_t})$             ▷ Sample $O_t$ from the categorical distribution $B_{\mathbf{x}_t}$
    $\mathbf{x}_{t+1} \sim \mathrm{Cat}(A_{\mathbf{x}_t})$        ▷ Sample $\mathbf{x}_{t+1}$ from the categorical distribution $A_{\mathbf{x}_t}$
end for
```

5.3 Three Problems

There are the three classic problems for traditional Hidden Markov Models:

Problem 1. Given $\boldsymbol{O}$ and λ, determine $\mathbb{P}(\boldsymbol{O}|\lambda)$.

Problem 2. Given $\boldsymbol{O}$ and λ, determine the most likely state sequence $\mathbf{X}$ to have generated $\boldsymbol{O}$, i.e. uncover the "hidden" aspect of the process. We adopt the approach of finding the state sequence that maximizes the expected number of correct states.

Problem 3. Given $\boldsymbol{O}$ and the cardinality N of the finite state space, find the model λ^* which will maximize $\mathbb{P}(\boldsymbol{O}|\lambda)$, i.e. fit a model to the data.

There are efficient algorithms for all three of these problems [14], which will not be given here. Instead, we provide an improved version of these algorithms to model an even more general model: a *Bursting Hidden Markov Model* (BHMM).

5.4 Bursting Hidden Markov Models

In a BHMM, each state emits not just a single observation, but a *burst* of observations. The length of these bursts may take on any value in the nonnegative integers. More specifically, we suppose that we also have a sequence of random variables $Y = \{Y_1, Y_2, \cdots\}$ taking values among the nonnegative integers. These may be drawn from any distribution over these values, but in this work we assume they result from Poisson parameters $\boldsymbol{\eta} = (\eta_i)$, where η_i denotes the average number of observations expected in a burst, given state i. Thus, we have

$$\mathbb{P}(T_t = k|\mathbf{x}_t = i) = \frac{\eta_i^k \cdot e^{-\eta_i}}{k!} \tag{5.4}$$

where T_t denotes the number of observations at time t.

A BHMM's parameters are now denoted

$$\lambda = \{A, B, \boldsymbol{\pi}, \boldsymbol{\eta}\}.$$

At each time, observations come in bursts, so instead of having an observation sequence, we now have a *burst sequence* $\boldsymbol{O} = (\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots)$ where

$$\boldsymbol{O}_t = (O_{t,1}, O_{t,2}, \cdots, O_{t,T_t})$$

and where $O_{t,i}$ is the i^{th} observation in the t^{th} burst, and T_t is the number of observations in the t^{th} burst. That is, each $O_{t,i}$ is drawn from the categorical distribution with parameters $B_{\mathbf{x}_t}$ and T_t is drawn from the Poisson distribution with parameter $\eta_{\mathbf{x}_t}$. Thus, graphically this is depicted with plate notation as in Figure 5.1.

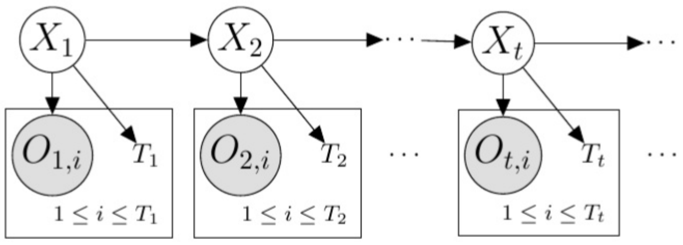

Figure 5.1: BHMM graphical plate notation.

We will throughout write $O_t(i)$ in place of $O_{t,i}$ for ease of visualization. This process will probably be best understood again by the generative algorithm given in Algorithm 5.2.

The question that remains is whether or not we can solve the same three problems with this new model. The answer is yes, but before showing this, we introduce notation to

Algorithm 5.2 Given the BHMM λ, generate an burst sequence $\mathbf{O}$

```
$\mathbf{x}_1 \sim \text{Cat}(\boldsymbol{\pi})$    ▷ Sample $\mathbf{x}_1$ from the initial state distribution $\boldsymbol{\pi}$
for $t = 1 \to T$ do
    $T_t \sim \text{Poiss}(\eta_{\mathbf{x}_t})$    ▷ Sample $T_t$ according to Poisson parameter $\eta_{\mathbf{x}_t}$
    for $s = 1 \to T_t$ do
        $O_t(s) \sim \text{Cat}(B_{\mathbf{x}_t})$    ▷ Sample $O_t(s)$ from the categorical distribution $B_{\mathbf{x}_t}$
    end for
    $\mathbf{x}_{t+1} \sim \text{Cat}(A_{\mathbf{x}_t})$    ▷ Sample $\mathbf{x}_{t+1}$ from the categorical distribution $A_{\mathbf{x}_t}$
end for
```

solidify our work up until this point.

5.5 Notation

Let us summarize our notation:

$$
\begin{aligned}
&N = \text{ number of states in the model}\\
&M = \text{ number of observation symbols}\\
&T = \text{ length of burst sequence}\\
&T_t = \text{ length of } t^{th} \text{ burst}\\
&Q = \{1, 2, \cdots, N\} = \text{ distinct states of the Markov process}\\
&S = \{1, 2, \cdots, M\} = \text{ set of possible observations}\\
&A = \text{ state transition probabilities } (N \times N \text{ row stochastic matrix})\\
&B = \text{ observation probabilities } (N \times M \text{ row stochastic matrix})\\
&\boldsymbol{\eta} = (\eta_i) = \text{ list of } N \text{ state-associated Poisson parameters}\\
&\boldsymbol{\pi} = (\pi_i) = \text{ initial state distribution (stochastic vector of length } N)\\
&\mathbf{O} = (\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_T) = \text{burst sequence}\\
&\mathbf{O}_t = (O_j(1), O_j(2), \cdots, O_j(T_t)) = t^{th} \text{ burst of observations}\\
&\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T) = \text{ state sequence.}
\end{aligned}
$$

5.6 Solutions to the Three Problems

Our solutions for BHMMs are a natural and intuitive extension of the solutions for the standard HMM found in [14]. We provide the solutions first, giving pseudocode for their implementation. Little justification is given here for the solution of the third problem, as this is fully given in the following section.

5.6.1 Solution to Problem 1. The solution of the first problem for a standard HMM uses a recursive algorithm known as the α-pass. Here we modify the α-pass for BHMMs. First, letting $\mathbf{O}$ be a burst sequence with $\mathbf{X}$ the corresponding state sequence, define a function

$$R(i,t) = \frac{\eta_i^{T_t} \cdot e^{-\eta_i}}{T_t!} \cdot b_{i,O_t(1)} \cdots b_{i,O_t(T_t)}. \tag{5.5}$$

Thus

$$\begin{aligned} R(i,t) &= \mathbb{P}(T_t|\mathbf{x}_t = i, \lambda) \cdot \mathbb{P}(O_t(1)|\mathbf{x}_t = i, \lambda) \cdots \mathbb{P}(O_t(T_t)|\mathbf{x}_t = i, \lambda) && (5.6)\\ &= \mathbb{P}(\mathbf{O}_t|\mathbf{x}_t = i, \lambda). && (5.7)\end{aligned}$$

Then in general we have

$$\begin{aligned} \mathbb{P}(\mathbf{O}|\mathbf{X}, \lambda) &= \mathbb{P}(\mathbf{O}_1|\mathbf{x}_1, \lambda) \cdot \mathbb{P}(\mathbf{O}_2|\mathbf{x}_2, \lambda) \cdots \mathbb{P}(\mathbf{O}_T|\mathbf{x}_T, \lambda) && (5.8)\\ &= R(\mathbf{x}_1, 1) \cdot R(\mathbf{x}_2, 2) \cdots R(\mathbf{x}_T, T) && (5.9)\end{aligned}$$

From the definition of $\boldsymbol{\pi}$ and A, we conclude that

$$\mathbb{P}(\mathbf{X}|\lambda) = \pi_{\mathbf{x}_1} a_{\mathbf{x}_1,\mathbf{x}_2} a_{\mathbf{x}_2,\mathbf{x}_3} \cdots a_{\mathbf{x}_{T-1},\mathbf{x}_T}. \tag{5.10}$$

By Bayes' Rule

$$\mathbb{P}(\mathbf{O}, \mathbf{X}|\lambda) = \mathbb{P}(\mathbf{O}|\mathbf{X}, \lambda) \cdot \mathbb{P}(\mathbf{X}|\lambda).$$

Since $\mathbb{P}(\mathbf{O}|\lambda) = \sum_{\mathbf{X}} \mathbb{P}(\mathbf{O}, \mathbf{X}|\lambda)$, we conclude that

$$\begin{aligned} \mathbb{P}(\mathbf{O}|\lambda) &= \sum_{\mathbf{X}} \mathbb{P}(\mathbf{O}|\mathbf{X}, \lambda) \cdot \mathbb{P}(\mathbf{X}|\lambda) && (5.11)\\ &= \sum_{\mathbf{X}} \pi_{\mathbf{x}_1} \cdot R(\mathbf{x}_1, 1) \cdot a_{\mathbf{x}_1,\mathbf{x}_2} \cdot R(\mathbf{x}_2, 2) \cdot a_{\mathbf{x}_2,\mathbf{x}_3} \cdot R(\mathbf{x}_3, 3) \cdots a_{\mathbf{x}_{T-1},\mathbf{x}_T} \cdot R(\mathbf{x}_T, T). && (5.12)\end{aligned}$$

The computation for $R(\mathbf{x}_t, t)$ requires $T_t + 1$ multiplications, so the above computation requires about $2T \cdot \sum_{t=1}^{T}(T_t + 1)$ multiplications, which is typically infeasible. Luckily the α-pass uses recursion to calculate this with much fewer multiplications. Define a $T \times N$ matrix α, where

$$\alpha_{t,i} = \mathbb{P}(\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_t, \mathbf{x}_t = i|\lambda). \tag{5.13}$$

Thus $\alpha_{t,i}$ is the probability of the partial burst sequence up to time t, where the Markov process is in state i at time t. For sake of appearance, we will write $\alpha_t(i)$ for $\alpha_{t,i}$. We can compute $\alpha_1(i)$ easily for each $i = 1, 2, \cdots, N$ as follows:

$$\begin{aligned}\alpha_1(i) &= \mathbb{P}(\mathbf{O}_1, \mathbf{x}_1 = i|\lambda) \\ &= \mathbb{P}(\mathbf{x}_1 = i|\lambda) \cdot \mathbb{P}(\mathbf{O}_1|\mathbf{x}_1 = i, \lambda) \\ &= \pi_i \cdot R(i, 1).\end{aligned}$$

Given $\alpha_{t-1}(i)$ we can also compute $\alpha_{t,i}$ as follows:

$$\begin{aligned}\alpha_t(i) &= \mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_t, \mathbf{x}_t = i|\lambda) \\ &= \mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_{t-1}, \mathbf{x}_t = i|\lambda) \cdot \mathbb{P}(\mathbf{O}_t|\mathbf{x}_t = i, \lambda) \\ &= \left[\sum_{j=1}^{N} \mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_{t-1}, \mathbf{x}_{t-1} = j, \mathbf{x}_t = i|\lambda)\right] \cdot R(i, t) \\ &= \left[\sum_{j=1}^{N} \mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_{t-1}, \mathbf{x}_{t-1} = j|\lambda) \cdot \mathbb{P}(\mathbf{x}_t = i|\mathbf{x}_{t-1} = j, \lambda)\right] \cdot R(i, t) \\ &= \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] \cdot R(i, t)\end{aligned}$$

Finally, note that

$$\begin{aligned}\mathbb{P}(\mathbf{O}|\lambda) &= \sum_{i=1}^{N} \mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_T, \mathbf{x}_T = i|\lambda)\\ &= \sum_{i=1}^{N} \alpha_T(i).\end{aligned}$$

We summarize this process by Algorithm 5.3.

Algorithm 5.3 Given λ and $\mathbf{O}$, find $\mathbb{P}(\mathbf{O}|\lambda)$

```
for $i = 1 \to N$ do
    $\alpha_1(i) = \pi_i \cdot R(i,1)$
end for
for $t = 2 \to T$ do
    for $i = 1 \to N$ do
        $\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] \cdot R(i,t)$
    end for
end for
$\mathbb{P}(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$
```

5.6.2 Solution to Problem 2.

We now use a modified β-pass to maximize the expected number of correct states in a state sequence, given a model λ and a burst sequence $\mathbf{O}$. Define another $T \times N$ matrix β where

$$\beta_{t,i} = \mathbb{P}(\mathbf{O}_{t+1}, \mathbf{O}_{t+2}, \cdots, \mathbf{O}_T|\mathbf{x}_t = i, \lambda), \tag{5.14}$$

again writing $\beta_t(i) = \beta_{t,i}$ for sake of clarity. Thus $\beta_T(i) = 1$ for all i. We can now recursively compute β by a "backward algorithm".

First, note that given $\beta_{t+1}(k)$ for $k = 1, 2, \cdots, N$ we can find $\beta_t(i)$ as follows:

$$
\begin{aligned}
\beta_t(i) &= \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T | \mathbf{x}_t = i, \lambda) \\
&= \sum_{j=1}^{N} \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T, \mathbf{x}_{t+1} = j | \mathbf{x}_t = i, \lambda) \\
&= \sum_{j=1}^{N} \mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_t = i, \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T | \mathbf{x}_{t+1} = j, \lambda) \\
&= \sum_{j=1}^{N} \mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_t = i, \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+1} | \mathbf{x}_{t+1} = j, \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+2}, \cdots, \mathbf{O}_T | \mathbf{x}_{t+1} = j, \lambda) \\
&= \sum_{j=1}^{N} a_{ij} \cdot R(j, t+1) \cdot \beta_{t+1}(j).
\end{aligned}
$$

Now, defining another $T \times N$ matrix γ by

$$
\begin{aligned}
\gamma_{t,i} &= \mathbb{P}(\mathbf{x}_t = i | \mathbf{O}, \lambda) \\
&= \frac{\mathbb{P}(\mathbf{x}_t = i, \mathbf{O} | \lambda)}{\mathbb{P}(\mathbf{O} | \lambda)} \\
&= \frac{\mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_t, \mathbf{x}_t = i | \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T | \mathbf{x}_t = i, \lambda)}{\mathbb{P}(\mathbf{O} | \lambda)} \\
&= \frac{\alpha_t(i) \cdot \beta_t(i)}{\mathbb{P}(\mathbf{O} | \lambda)}.
\end{aligned}
$$

we have for a given burst sequence $\mathbf{O}$ and model λ, the most likely state at time t is

$$
\mathbf{x}_t^* = \operatorname*{argmax}_i \gamma_t(i) \tag{5.15}
$$

We summarize this process by Algorithm 5.4:

5.6.3 Solution to Problem 3. Problem 3 is certainly the most difficult to solve, but also provides the greatest power to HMMs and BHMMs. Here, we fit a model to a burst sequence. In this case, suppose that we know N and M, and we are also given a burst sequence $\mathbf{O}$. Our objective is to maximize $\mathbb{P}(\mathbf{O}|\lambda)$ subject to the cardinality constraints

Algorithm 5.4 Given λ and $\mathbf{O}$, find the most likely state sequence $\mathbf{X}$ to have generated $\mathbf{O}$.

```
Compute $\alpha$ by Algorithm 5.3
for $i = 1 \to N$ do
    $\beta_T(i) = 1$
end for
for $t = T - 1 \to 1$ do
    for $i = 1 \to N$ do
        $\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot R(j, t+1) \cdot \beta_{t+1}(j)$
    end for
end for
for $t = 1 \to T$ do
    $\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{\mathbb{P}(\mathbf{O}|\lambda)}$
    $\mathbf{x}_t^* = \text{argmax}_i \, \gamma_t(i)$
end for
```

of the state and observation spaces. To begin, we must initialize our model λ with our best guess. If we do not have any educated guess available, choose random values so that $\pi_i \approx \frac{1}{N}, a_{ij} \approx \frac{1}{N}$, and $b_{ij} \approx \frac{1}{M}$, preserving the appropriate stochasticities. Given our current model λ, we seek to update to a model $\widehat{\lambda}$ such that $\mathbb{P}(\mathbf{O}|\widehat{\lambda}) \geq \mathbb{P}(\mathbf{O}|\lambda)$. We iterate this process until we meet some convergence tolerance τ. Now, given only the dimensions of our spaces and our burst sequence, we need to reestimate $A, B, \boldsymbol{\pi}$ and $\boldsymbol{\eta}$. Unfortunately, to do this we must further define a three dimensional tensor δ of dimensions $T - 1 \times N \times N$, where

$$\delta_{t,i,j} = \mathbb{P}(\mathbf{x}_t = i, \mathbf{x}_{t+1} = j | \mathbf{O}, \lambda). \tag{5.16}$$

Thus, $\delta_{t,i,j}$ is the probability of being in state i at time t and transitioning to state j, given the model and the observations, and once again, we write $\delta_t(i, j)$ for $\delta_{t,i,j}$. We can compute this as follows:

$$\begin{aligned}
\delta_t(i,j) &= \mathbb{P}(\mathbf{x}_t = i, \mathbf{x}_{t+1} = j | \mathbf{O}, \lambda) \\
&= \frac{\mathbb{P}(\mathbf{x}_t = i, \mathbf{x}_{t+1} = j, \mathbf{O} | \lambda)}{\mathbb{P}(\mathbf{O}|\lambda)} \\
&= \frac{\mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_t, \mathbf{x}_t = i, \mathbf{x}_{t+1} = j|\lambda) \cdot \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T | \mathbf{x}_{t+1} = j, \lambda)}{\mathbb{P}(\mathbf{O}|\lambda)} \\
&= \frac{\mathbb{P}(\mathbf{O}_1, \cdots, \mathbf{O}_t, \mathbf{x}_t = i|\lambda) \cdot \mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_t = i, \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+1}, \cdots, \mathbf{O}_T | \mathbf{x}_{t+1} = j, \lambda)}{\mathbb{P}(\mathbf{O}|\lambda)} \\
&= \frac{\alpha_t(i) \cdot a_{ij} \cdot \mathbb{P}(\mathbf{O}_{t+1} | \mathbf{x}_{t+1} = j, \lambda) \cdot \mathbb{P}(\mathbf{O}_{t+2}, \cdots, \mathbf{O}_T | \mathbf{x}_{t+1} = j)}{\mathbb{P}(\mathbf{O}|\lambda)} \\
&= \frac{\alpha_t(i) \cdot a_{ij} \cdot R(j, t+1) \cdot \beta_{t+1}(j)}{\mathbb{P}(\mathbf{O}|\lambda)}
\end{aligned}$$

Now, using the matrix γ and our tensor δ, and some other readily available information, we can reestimate our model. We first consider the task of reestimating our transitions probability matrix A.

Reestimation of A**.** To reestimate A given our burst sequence $\mathbf{O}$, for each $i, j = 1, 2, \cdots, N$ we must find the expected number of transitions from i to j, as well as the total expected number of transitions from i to any state. The first expectation is $\sum_{t=1}^{T-1} \delta_t(i,j)$ and the second expectation is $\sum_{t=1}^{T-1} \gamma_t(i)$. Thus our reestimation formula is

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \delta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{5.17}$$

Reestimation of B**.** To reestimate B given our burst sequence $\mathbf{O}$, for each $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, N$ we must find the expected number of times observation j is emitted while in state i, as well as the total expected number of observations emitted while in state i. The first expectation is $\sum_{t=1}^{T} \gamma_t(i) \cdot C(t,j)$ where $C(t,j)$ is the number of occurrences of observation j in the t^{th} burst $\mathbf{O}_t$. The second expectation is $\sum_{t=1}^{T} \gamma_t(i) \cdot T_t$. Thus our reestimation formula is

$$\widehat{b}_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot C(t,j)}{\sum_{t=1}^{T} \gamma_t(i) \cdot T_t}. \tag{5.18}$$

Reestimation of $\boldsymbol{\pi}$. Reestimation of $\boldsymbol{\pi}$ is quite simple, since $\pi_i = \mathbb{P}(\mathbf{x}_1 = i|\lambda) = \gamma_1(i)$. Thus the reestimation formula is

$$\widehat{\pi}_i = \gamma_1(i). \tag{5.19}$$

Reestimation of $\boldsymbol{\eta}$. To reestimate $\boldsymbol{\eta}$ given our burst sequence $\mathbf{O}$, for each $i = 1, 2, \cdots, N$ we must find the expected number of observations emitted while in state i as well as the expected number of occurrences of state i. These expectations have been computed above, and our reestimation formula is

$$\widehat{\eta}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot T_t}{\sum_{t=1}^{T} \gamma_t(i)}. \tag{5.20}$$

We re-estimate iteratively, each iteration hopefully increasing the probability of the burst sequence given the updated model. We summarize this process with Algorithm 5.5. In practice, some computational issues must be dealt with pertaining to underflow. See [14] for dealing with underflow in a standard HMM. Extending these techniques to BHMMs is fairly straightforward.

The reader may be wondering why this iterative process improves our estimation of λ, and has probably already made the connection that its success is a direct result of Expectation-Maximization. We now demonstrate this.

Algorithm 5.5 Given $\mathbf{O}$, N, M, and some tolerance τ, find the model λ most likely to have generated $\mathbf{O}$.

```
Initialize $\lambda = (A, B, \boldsymbol{\pi}, \boldsymbol{\eta})$
Count = 0.
Compute $\alpha$ and $\mathbb{P}(\mathbf{O}|\lambda)$ by Algorithm 5.3
while Count < MaxIters do
    Count = Count +1
    Compute $\beta$ and $\gamma$ by Algorithm 5.4
    for $t = 1 \to T-1$ do
        for $i = 1 \to N$ do
            for $i = 1 \to N$ do
                $\delta_t(i,j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot R(j,t+1) \cdot \beta_{t+1}(j)}{\mathbb{P}(\mathbf{O}|\lambda)}$
            end for
        end for
    end for
    for $i = 1 \to N$ do
        for $j = 1 \to N$ do
            $\widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \delta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$
        end for
        for $j = 1 \to M$ do
            $\widehat{b}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i) \cdot C(t,j)}{\sum_{t=1}^{T-1} \gamma_t(i) \cdot T_t}$
        end for
        $\widehat{\pi}_i = \gamma_1(i)$
        $\widehat{\eta}_i = \frac{\sum_{t=1}^{T-1} \gamma_t(i) \cdot T_t}{\sum_{t=1}^{T-1} \gamma_t(i)}$
    end for
    $\widehat{\lambda} = (\widehat{A}, \widehat{B}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\eta}})$
    Compute $\alpha$ and $\mathbb{P}(\mathbf{O}|\widehat{\lambda})$ by Algorithm 5.3
    if $\mathbb{P}(\mathbf{O}|\widehat{\lambda}) - \mathbb{P}(\mathbf{O}|\lambda) < \tau$ then return $\widehat{\lambda}$
    end if
    Update $\lambda = \widehat{\lambda}$
end while
return $\lambda$
```

5.7 Reestimation as Expectation-Maximization

The solution to problem 3 is exactly the Expectation-Maximization algorithm. A nice formulation of the solution to problem 3 for standard HMMs can be found in [15]. To see this formulation for BHMMs, we consider EM as a tool to estimate λ given $\mathbf{O}$. Recall that in EM, we seek to find λ^* which maximizes $L(\lambda; \mathbf{O}) = \mathbb{P}(\mathbf{O}|\lambda)$, where $\mathbf{O}$ is known data. To do this, we start with an initialization λ_0. Given a parameter estimate λ_n, we compute the conditional expectation of the log-likelihood of λ given $\mathbf{O}$ and our unknown variables $\mathbf{X}$, conditioned on $\mathbf{O}, \lambda_n$, i.e. we compute

$$\begin{aligned} Q(\lambda|\lambda_n) &= E_{\mathbf{X}|\mathbf{O},\lambda_n}\left[\ln \mathbb{P}(\mathbf{O}, \mathbf{X}|\lambda)\right] \\ &= \sum_{\mathbf{X}} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \ln \mathbb{P}(\mathbf{O}, \mathbf{X}|\lambda). \end{aligned}$$

We then maximize $Q(\lambda|\lambda_n)$ with respect to λ, and set $\lambda_{n+1} = \operatorname*{argmax}_{\lambda} Q(\lambda|\lambda_n)$.

Lemma 5.1. *Given a parameter estimate λ_n, the negative of our conditional expectectaion $-Q(\lambda|\lambda_n)$ is convex.*

Proof. Suppose we have a current estimate of our parameters λ_n. If we define $p(\mathbf{X}) =$

$\mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)$, then we can compute

$$\begin{aligned}
-Q(\lambda|\lambda_n) &= -\sum_{\mathbf{X}} p(\mathbf{X}) \ln \mathbb{P}(\mathbf{O}, \mathbf{X}|\lambda) \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \ln \left[\mathbb{P}(\mathbf{O}|\mathbf{X}, \lambda)\mathbb{P}(\mathbf{X}|\lambda)\right] \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \ln \left[\left(\prod_{t=1}^{T} \mathbb{P}(\mathbf{O}|\mathbf{x}_t, B, \boldsymbol{\eta})\right)\left(\mathbb{P}(\mathbf{x}_1|\boldsymbol{\pi}) \prod_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_{t+1}|\mathbf{x}_t, A)\right)\right] \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \ln \left[\left(\prod_{t=1}^{T} \mathbb{P}(T_t|\mathbf{x}_t, \boldsymbol{\eta}) \prod_{s=1}^{T_t} \mathbb{P}(O_t(s)|\mathbf{x}_t, B)\right)\left(\mathbb{P}(\mathbf{x}_1|\boldsymbol{\pi}) \prod_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_{t+1}|\mathbf{x}_t, A)\right)\right] \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \ln \left[\left(\prod_{t=1}^{T} \frac{\eta_{\mathbf{x}_t}^{T_t} \cdot e^{-\eta_{\mathbf{x}_t}}}{T_t!} \prod_{s=1}^{T_t} b_{\mathbf{x}_t, O_t(s)}\right)\left(\pi_{x_1} \prod_{t=1}^{T-1} a_{x_t, x_{t+1}}\right)\right] \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \left[\sum_{t=1}^{T}\left(\ln \frac{\eta_{\mathbf{x}_t}^{T_t} \cdot e^{-\eta_{\mathbf{x}_t}}}{T_t!} + \ln a_{\mathbf{x}_t, \mathbf{x}_{t+1}} + \sum_{s=1}^{T_t} \ln b_{\mathbf{x}_t, O_t(s)}\right) - \ln a_{\mathbf{x}_T, \mathbf{x}_{T+1}} + \ln \pi_{\mathbf{x}_1}\right] \\
&= -\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{t=1}^{T}\left[\mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t=i \wedge t \neq T]} \ln a_{ij} + \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]} \ln b_{ik}\right.\right. \\
&\qquad\qquad \left.\left. + \ \mathbb{1}_{[\mathbf{x}_t=i]} \ln \frac{\eta_i^{T_t} \cdot e^{-\eta_i}}{T_t!}\right] + \mathbb{1}_{[\mathbf{x}_1=i]} \ln \pi_i\right) \\
&= \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{t=1}^{T}\left[\mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t=i \wedge t \neq T]}(-\ln a_{ij}) + \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]}(-\ln b_{ik})\right.\right. \\
&\qquad\qquad \left.\left. + \ \mathbb{1}_{[\mathbf{x}_t=i]}(T_t(-\ln \eta_i) + \ln T_t! + \eta_i)\right] + \mathbb{1}_{[\mathbf{x}_1=i]}(-\ln \pi_i)\right)
\end{aligned}$$

Note that Q is a function of $\lambda = (A, B, \boldsymbol{\pi}, \boldsymbol{\eta})$, so our "variables" are the a_{ij}, b_{ik}, η_i, and π_i. Furthermore, $p(\mathbf{X}) = \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)$ is not a function of λ, and is thus a nonnegative constant, as is T_t. Recall that $-\ln(x)$ is a convex function of x, so here, $-\ln(a_{ij}), -\ln(b_{ik}), -\ln(\eta_i)$, and $-\ln(\pi_i)$ are convex functions of our variables. Recall also that linear functions are convex (hence $\eta_i + \ln T_t!$ is convex), a nonnegative multiple of a convex function is convex, and the sum of convex functions is convex. Thus, we have here a sum of convex functions, hence $-Q(\lambda|\lambda_n)$ is convex.

$\square$

Theorem 5.2. *The reestimations of $A, B, \boldsymbol{\pi}$, and $\boldsymbol{\eta}$ as given in Equations 5.17 - 5.20 are the* *reestimations given by the Expectation-Maximization algorithm for Bursting Hidden Markov Models.*

Proof. We have just calculated that the E-step of the algorithm yields

$$Q(\lambda|\lambda_n) = \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \sum_{k=1}^{M} \sum_{t=1}^{T} \left[\mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i \wedge t \neq T]} \ln a_{ij} + \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t = i]} \ln b_{ik} \right.\right.$$
$$\left.\left. + \mathbb{1}_{[\mathbf{x}_t = i]} (T_t \ln \eta_i - \ln T_t! - \eta_i) \right] + \mathbb{1}_{[\mathbf{x}_1 = i]} \ln \pi_i \right)$$

The M-step requires us to find the argmax of $Q(\lambda|\lambda_n)$ with respect to λ, which is equivalent to solving the following convex program:

$$\begin{aligned} \text{Minimize:} \quad & -Q(\lambda|\lambda_n) \\ \text{Subject To:} \quad & 1 - \sum_{j=1}^{N} a_{ij} = 0 \text{ for each } 1 \leq i \leq N \\ & 1 - \sum_{k=1}^{M} b_{ik} = 0 \text{ for each } 1 \leq i \leq N \\ & 1 - \sum_{i=1}^{N} \pi_i = 0 \end{aligned}$$

Note that we have ignored the nonnegative constraints on our variables a_{ij}, b_{ij}, π_i and η_i. This will shortly be justified when we see that the solution to the Karush-Kuhn-Tucker conditions [4] already satisfies those constraints. Setting up the Lagrangian $\mathcal{L} = \mathcal{L}(A, B, \boldsymbol{\pi}, \boldsymbol{\eta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \gamma)$ for this optimization problem gives us

$$\mathcal{L} = \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \sum_{k=1}^{M} \sum_{t=1}^{T} \left[\mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i \wedge t \neq T]} (-\ln a_{ij}) + \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t = i]} (-\ln b_{ik}) \right.\right.$$
$$\left.\left. + \mathbb{1}_{[\mathbf{x}_t = i]} (-T_t \ln \eta_i + \ln T_t! + \eta_i) \right] + \mathbb{1}_{[x_1 = i]} (-\ln \pi_i) \right)$$
$$+ \sum_{i=1}^{N} \delta_i (1 - \sum_{j=1}^{N} a_{ij}) + \sum_{i=1}^{N} \epsilon_i (1 - \sum_{k=1}^{M} b_{ik}) + \gamma (1 - \sum_{i=1}^{N} \pi_i)$$

We seek a solution to the KKT conditions to find the argmax. Taking the partial derivative with respect to a_{ij} and setting it equal to zero, we have

$$\frac{\partial \mathcal{L}}{\partial a_{ij}} = \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i \wedge t \neq T]} \frac{-1}{a_{ij}} - \delta_i = 0 \tag{5.21}$$

$$\Longrightarrow a_{ij} = -\frac{1}{\delta_i} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i]} \tag{5.22}$$

Since $1 - \sum_{j=1}^{N} a_{ij} = 0$, this then becomes

$$0 = 1 + \sum_{j=1}^{N} \frac{1}{\delta_i} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i]} \tag{5.23}$$

$$\Longrightarrow \delta_i = -\sum_{j=1}^{N} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i]} \tag{5.24}$$

$$= -\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_t = i]}. \tag{5.25}$$

Combining 5.22 and 5.25, we reestimate a_{ij} as

$$\begin{aligned}
\widehat{a_{ij}} &= \frac{-\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i]}}{-\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T-1} \mathbb{1}_{[\mathbf{x}_t = i]}} \\
&= \frac{\sum_{t=1}^{T-1} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_{t+1}=j \wedge \mathbf{x}_t = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)}{\sum_{t=1}^{T-1} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_t = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)} \\
&= \frac{\sum_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_{t+1} = j, \mathbf{x}_t = i|\mathbf{O}, \lambda_n)}{\sum_{t=1}^{T-1} \mathbb{P}(\mathbf{x}_t = i|\mathbf{O}, \lambda_n)} \\
&= \frac{\sum_{t=1}^{T-1} \delta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}
\end{aligned}$$

which is precisely our reestimate for a_{ij} as given in 5.17.

Similarly, we compute the partial derivative of $\mathcal{L}$ with respect to b_{ik} and set it equal to zero:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial b_{ik}} &= \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]} \frac{-1}{b_{ik}} - \epsilon_i = 0 \\ \implies b_{ik} &= -\frac{1}{\epsilon_i} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]}\end{aligned}$$

Since $1 - \sum_{k=1}^{M} b_{ik} = 0$, this then becomes

$$\begin{aligned}0 &= 1 + \sum_{k=1}^{M} \frac{1}{\epsilon_i} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]} \\ \implies \epsilon_i &= -\sum_{k=1}^{M} \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]} \\ &= -\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[\mathbf{x}_t=i]} \\ &= -\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[\mathbf{x}_t=i]} \cdot T_t\end{aligned}$$

Thus, letting $C(t,k)$ denote the number of occurences of observation k in the t^{th} burst, we have

$$\begin{aligned}\widehat{b_{ik}} &= \frac{-\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \sum_{s=1}^{T_t} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]}}{-\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[x_t=i]} \cdot T_t} \\ &= \frac{\sum_{t=1}^{T} \sum_{s=1}^{T_t} \sum_{\mathbf{X}} \mathbb{1}_{[O_t(s)=k \wedge \mathbf{x}_t=i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)}{\sum_{t=1}^{T} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_t=i]} \cdot \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \cdot T_t} \\ &= \frac{\sum_{t=1}^{T} \mathbb{P}(\mathbf{x}_t = i|\mathbf{O}, \lambda_n) \cdot C(t,k)}{\sum_{t=1}^{T} \mathbb{P}(\mathbf{x}_t = i|\mathbf{O}, \lambda_n) \cdot T_t} \\ &= \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot C(t,k)}{\sum_{t=1}^{T} \gamma_t(i) \cdot T_t}\end{aligned}$$

which is again, precisely our reestimate for b_{ik} as in 5.18.

We now examine the reestimate for $\boldsymbol{\pi}$:

$$\frac{\partial \mathcal{L}}{\partial \pi_i} = \sum_{\mathbf{X}} p(\mathbf{X}) \mathbb{1}_{[\mathbf{x}_1 = i]} \frac{-1}{\pi_i} - \gamma = 0$$
$$\implies \pi_i = -\frac{1}{\gamma} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_1 = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)$$

Since $1 - \sum_{i=1}^{N} \pi_i = 0$, we have

$$\begin{aligned} 0 &= 1 + \sum_{i=1}^{N} \frac{1}{\gamma} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_1 = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \\ \implies \gamma &= -\sum_{i=1}^{N} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_1 = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \\ &= -\sum_{\mathbf{X}} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \\ &= -1 \end{aligned}$$

and hence we reestimate π_i as

$$\begin{aligned} \widehat{\pi_i} &= \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_1 = i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \\ &= \mathbb{P}(\mathbf{x}_1 = i|\mathbf{O}, \lambda_n) \\ &= \gamma_1(i), \end{aligned}$$

our reestimation for π_i provided in 5.19.

It remains to show that our reestimate for η_i as given in 5.20 is accurate:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \eta_i} &= \sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[\mathbf{x}_t=i]}(\frac{-T_t}{\eta_i} + 1) = 0 \\
\implies \widehat{\eta_i} &= \frac{\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[\mathbf{x}_t=i]} T_t}{\sum_{\mathbf{X}} p(\mathbf{X}) \sum_{t=1}^{T} \mathbb{1}_{[\mathbf{x}_t=i]}} \\
&= \frac{\sum_{t=1}^{T} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_t=i]} \cdot \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n) \cdot T_t}{\sum_{t=1}^{T} \sum_{\mathbf{X}} \mathbb{1}_{[\mathbf{x}_t=i]} \mathbb{P}(\mathbf{X}|\mathbf{O}, \lambda_n)} \\
&= \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot T_t}{\sum_{t=1}^{T} \gamma_t(i)}
\end{aligned}$$

which is our reestimation for η_i given in 5.20.

Since this is a solution to the KKT conditions, we have shown that

$$\underset{\lambda}{\operatorname{argmax}}\, Q(\lambda|\lambda_n) = \widehat{\lambda} = (\widehat{A}, \widehat{B}, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\eta}})$$

as given in Equations 5.17 - 5.20. □

Corollary 5.3. *The solution provided for Problem 3 is an implementation of Expectation-Maximization for HMMs.*

Proof. This is immediate from the theorem. □

Chapter 6. Data Driven Diagnosis of Chronic Kidney Disease

6.1 Introduction

When a chronic disease is detected early, it is often more susceptible to medical treatments, providing the victim a better quality of life for a longer period of time, while simultaneously decreasing medical costs. With this in mind, statistical learning tools targeting early disease detection not only have the potential of improving the lives of thousands of people each year, they also carry the ability to yield significant savings to insurance providers by allowing the prevention of a chronic disease from progressing to a late stage which is often difficult (if not impossible) to sucessfully treat. Chronic Kidney Disease (CKD) is certainly one such disease.

Such statistical learning models have been implemented with moderate success based on certain summary variables describing each patient. These variables may be somewhat standard of any study of a population, such as age and gender, but are more often specifically medical in nature, such as the Charlson index, the number of hospitalizations, or the number of distinct providers seen in a year. While these summary variables have been used well in a binary logistic regression to classify individuals into two categories (CKD or no CKD), a weakness of such a method is that it only considers a snapshot of a patient's medical history. Summary variables are just that - summaries. As such, they contain important information, yet fail to retain the information imbedded in the progression of a patient's health state.

We propose to improve existing models by incorporating progressional data. One such variable which is found in nearly every insurance claim is the diagnosis code. When a patient visits a healthcare provider and a procedure is performed or a prescription given, a medical diagnosis code is recorded indicating the reason of the visit. These codes can be symptomatic in nature (786.05 - "shortness of breath") or more disease specific (250.2 - "diabetes with hyperosmolarity"). While these codes also represent "snapshots" of a patient's medical history, they provide a more specific image of the health issues affecting a given patient. It is a reasonable assumption that patients who will soon develop CKD will likely be suffering from similar ailments, somewhat distinct from those afflicting those who never get CKD.

Unfortunately, there are well over 13,000 different ICD-9 diagnosis codes, and so a metric on this space may not only be difficult to define, but also a poor measure of similarity (or dissimilarity) of patients. We present a method to extract scaled-down information from patients' diagnosis codes via LDA.

6.2 LDA and Disease Detection

The use of LDA for disease detection serves as a tool to extract meaningful information from ICD-9 diagnosis codes. If we allow each ICD-9 diagnosis code to be a term in the vocabulary, each patient to be represented as a document, whose words are the specific instances of ICD-9 codes given to the patient during a set period of time, then any collection of patient data can be considered a corpus, and hence is susceptible to the methods of LDA. Therefore, taking a large sampling of patients and K, a set number of topics, we can infer a topic distribution for each patient, and a term distribution for each of the K topics. For patient d, each of the K entries in $\boldsymbol{\theta}_d$ provide added information about the patient's health state, yielding a total of K new predictor variables, based on the patient's distribution over the K health topics.

A few words of caution: if we choose K to be too large, then we will add too many predictor variables, likely forcing any trained classifier to overfit the data by increasing the model complexity. Also, if we do not preprocess the data to remove stop words, our topics will be filled with "junk" terms.

In natural language processing, a stop word is a term that will occur frequently in nearly every document of a corpus. In English text, examples of stop words include *the, a, an, and, if, to, this,* etc. These words will be among the most likely words in any topic, even though they provide no information about the topic. This problem is normally dealt with in text analysis by compiling a list of stop words and discarding them from each document prior to inferring an LDA model on the data. Similarly, some ICD-9 diagnosis codes occur so frequently over so many people that they are essentially meaningless in their predictive abilities. These include the following:

V70.0 –	Routine general medical examination.
786.50 –	Chest pain, unspecified.
V72.31 –	Routine gynecological examination.
780.79 –	Other malaise and fatigue.
462 –	Acute pharyngitis.
466.0 –	Acute bronchitis.
724.2 –	Lumbago (lower back pain).
V04.81 –	Influenza vaccination.

Others may occur so frequently that one might consider them to be stop words, though they still provide significant information about a patient's health issues. A primary example of this is Diabetes Mellitus, which is so prevalent that one might consider it a stop word, though it certainly is not. For ICD-9 diagnosis codes, a stop words list should be compiled initially by overall frequency, and then intelligently selected to avoid removing meaningful ICD-9 diagnosis codes from the corpus vocabulary.

6.3 BHMM and Disease Detection

We can also consider that patients may transition through various states of health throughout their lifetime. We can model these transitions through these unobservable states by an HMM where the observations are the diagnosis codes. Since it is unlikely that state transitions occur each time an insurance claim is submitted (which could be multiple times in a day), we aggregate diagnosis codes in three month periods. This allows us to use BHMMs to model the process, since in each three month interval we obtain a burst of data. We assume that the collection of diagnosis codes in each three month period is itself a document, instead of each patient's full collection of diagnosis codes being a full document. With this formulation, we can represent each patient by a burst sequence.

Once again, the observation space for this data is large, and thus difficult to model. We use LDA to reduce the dimensionality of the state space by assigning each diagnosis code for each patient to the topic from which it was most likely generated. More precisely, for each word $w_{a,b}$, we assign

$$z_{a,b}^{MLE} = \underset{k}{\operatorname{argmax}}\, \mathbb{P}(z_{a,b} = k | w_{a,b}, \widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\theta}}_b)$$

where $\widehat{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\theta}}_b$ are our estimates of $\boldsymbol{\Phi}$ and $\boldsymbol{\theta}_b$ respectively. Note that

$$\begin{aligned}
\mathbb{P}(z_{a,b} = k | w_{a,b}, \widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\theta}}_b) &= \frac{\mathbb{P}(z_{a,b} = k, w_{a,b} | \widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\theta}}_b)}{\mathbb{P}(w_{a,b} | \widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\theta}}_b)} \\
&= \frac{\mathbb{P}(w_{a,b} | z_{a,b} = k, \widehat{\boldsymbol{\Phi}}) \cdot \mathbb{P}(z_{a,b} = k | \widehat{\boldsymbol{\theta}}_b)}{\mathbb{P}(w_{a,b} | \widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\theta}}_b)} \\
&= \frac{\widehat{\phi_{k,w_{a,b}}} \cdot \widehat{\theta_{b,k}}}{\sum_{i=1}^{K} \widehat{\phi_{i,w_{a,b}}} \cdot \widehat{\theta_{b,i}}}
\end{aligned}$$

Thus we have that

$$z_{a,b}^{MLE} = \underset{k}{\operatorname{argmax}}\, \widehat{\phi_{k,w_{a,b}}} \cdot \widehat{\theta_{b,k}}.$$

With this hard topic assignment, we can reduce the dimension of the observation space for the BHMM from over 13,000 to some relatively small K, thereby making the training procedure for the BHMM feasible.

6.4 The Analysis

Our dataset we used for this analysis initially consisted of health insurance claims for approximately 485,000 patients from a Midwestern metropolitan area. From our claims dataset, we consider as target cases all patients who were diagnosed with CKD (were given an ICD-9 diagnosis code 403, 404, 584, or 585) and had diagnosis codes outside the stop word list a year or more before the beginning of the year in which they were first diagnosed. We consider as control cases all patients who were never diagnosed with CKD and had diagnosis codes outside the stop word list a year or more before the beginning of their final year of data. We refer to the year of diagnosis and the final year of data for CKD and control patients respectively as a patient's *final year*. After this restriction, we had 265,546 patients classified as "Control" and 2,280 patients classified as "CKD". The list of stopwords was obtained by validation testing, as explained below.

We segmented each patient's diagnosis codes into three month bursts for each year prior to the final year. Randomly sampling 500 patients of class CKD and 10,000 patients of class Control, we trained a BHMM on the sampled CKD patients and a separate BHMM on the sampled control patients as described in 6.3, using 30 topics in the LDA inference for the dimensionality reduction.

For each of the remaining 1,780 CKD patients and 255,546 control patients, we subtracted the log-likelihood that the patient's data was generated by the control BHMM from the log-likelihood that it came from the CKD BHMM and called these values the *scores*. We also obtained their age, gender, Charlson index, allowable amount charged to the insurance company, and the number of hospitalizations, each for the year prior to the patient's final year. We also collected all diagnosis codes for each patient prior to the final year, and treating each patient's collection of diagnosis codes as a document as in Section 6.2, we estimated the topic distribution for each patient using LDA inference. Thus each of the patients was represented by the following variables: score, age, gender, Charlson index, allowable amount, number of hospitalizations, and the components of the topic distribution. We separated the data into three sets: training, validation, and testing. The training set consisted of 1,000 randomly sampled CKD patients and 10,000 randomly sampled control patients. The validation set contained 280 CKD patients and 10,000 control patients, leaving 500 CKD patients and 235,546 control patients for testing.

We then trained random forests of 500 trees on the classifier training set with $p = \lfloor \sqrt{P} \rfloor$, where p and P are as given in Section 3.4, varying our five free parameters: the list of stop words, the number of topics, the symmetric hyperparameters α and β, and the number of states for the BHMMs. Using a greedy search with our validation testing, we settled on a specific list of stopwords, 60 topics, hyperparameters $\alpha = 0.1$ and $\beta = 0.1$, and 10 BHMM states. Of the parameter sets tested, these parameters yielded the greatest ROC score. Using these parameters, we also trained three additional random forests: the first used only the 5 summary variables, the second used only the summary variables and the scores, and the third used all the variables except the scores. Using each of these random forests, we estimated the probability of being class "CKD" or class "Control" for each patient in the classifier test set.

For each random forest, we computed various threshold values, allowing us to analyze the four classifiers with a ROC curve, as seen in Figure 6.1. In Table 6.1 we provide the ROC score for each classifier. As expected, the weakest predictor was the random forest with only the summary variables. Including the scores from the BHMMs provided some additional lift, but when including the topic distributions there was significant improvement. Interestingly, including the random forest trained on all variables did not perform significantly better than the random forest trained on all variables but the scores. This suggests a near total redundancy of information gained from the BHMMs and that gained from the LDA topic analysis, but also suggesting that overal topic distributions are better predictors than the Markov process transitioning through health states.

Classifier	ROC Score
Summary Variables	0.754
BHMM with Summary Variables	0.799
LDA with Summary Variables	0.835
BHMM and LDA with Summary Variables	0.838

Table 6.1: ROC scores for four random forests trained on the claims data.

In Tables 6.2 and 6.3 we provide specificity-sensitivity pairs for the best and worst classifiers, respectively. As the specificity decreases, the improvement from including data

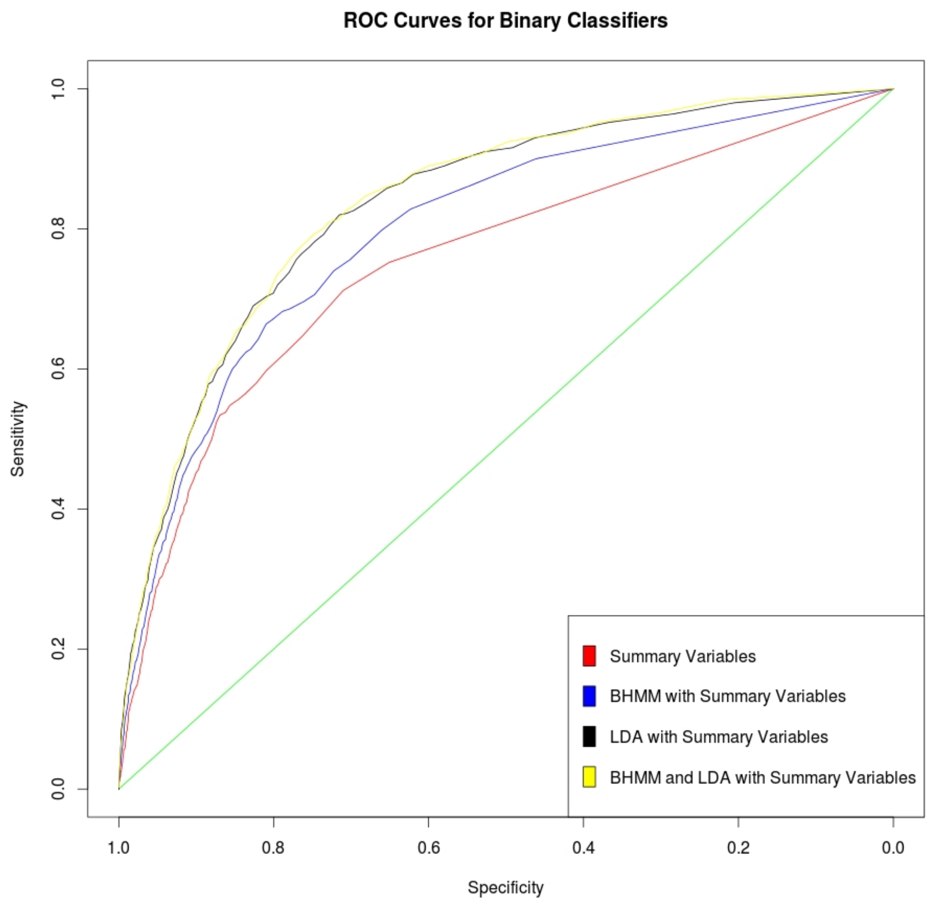

Figure 6.1: ROC curves for four random forests.

from the diagnosis codes tends to increase. The overall improvement in classification ability indicates that diagnosis code data, in addition to summary variable data, can play an important role in predictive disease diagnosis.

6.4.1 Policy Choices. Choosing the correct threshold at which to flag patients for further medical testing may seem subjective. It certainly depends on the amount of false positives deemed tolerable. A simple objective way to determine this choice is to create a simple cost-savings function on the specificity-sensitivity pairs and maximize this function. Given an estimated yearly savings of early CKD diagnosis for a patient and an estimated

Specificity	Sensitivity
99.5	8.6
99.0	14.4
98.0	21.0
97.1	26.8
95.2	36.4
90.0	53.0
84.9	65.2
80.0	73.4

Table 6.2: Specificity-Sensitivity pairs for random forest trained on summary variables with BHMM and LDA.

Specificity	Sensitivity
99.5	4.0
99.0	8.2
98.0	14.0
97.1	18.0
95.2	28.8
90.0	45.2
84.6	55.6
80.9	59.8

Table 6.3: Specificity-Sensitivity pairs for random forest trained on summary variables.

screening cost, this function could be as simple as

$$C(n_{\text{CKD}}, n_{\text{Screened}}) = CKD_{\text{cost}} \cdot n_{\text{CKD}} - S_{\text{cost}} \cdot n_{\text{Screened}}$$

where n_{CKD} is the number of people who would develop CKD in a year and are identified, n_{Screened} is the number of people needed to screen in order to detect those diagnosed with CKD, CKD_{cost} is the estimated yearly savings of early CKD diagnosis per patient, and S_{cost} is the estimated screening cost per patient. Thus supposing a population of size 301,500 with 1,500 new CKD diagnoses in a year where $CKD_{\text{cost}} = \$10{,}000$ and $S_{\text{cost}} = \$100$, for each of the above specificity-sensitivity pairs for our random forest with summary variables and LDA we have the estimated cost savings given in Table 6.4.

Then selecting the threshold yielding the specificity-sensitivity pair of 84.9% – 65.2% would give the greatest cost savings, at $5.15 million, whereas the greatest cost savings under

Specificity	Sensitivity	Savings
99.5	8.6	\$1.13M
99.0	14.4	\$1.84M
98.0	21.0	\$2.52M
97.1	26.8	\$3.11M
95.2	36.4	\$3.97M
90.0	53.0	\$4.87M
84.9	65.2	\$5.15M
80.0	73.4	\$4.90M

Table 6.4: Example of cost savings estimates for the random forest trained on summary variables with BHMM and LDA.

the model trained just on summary variables yields \$3.71 million, a potentially significant difference for a population of this size. In this manner, policy may be determined objectively to provide the greatest cost savings, and consequentially to significantly improve the lives of over 65% of those likely to develop CKD sometime within the next year.

Chapter 7. Conclusion

Our particular solution to predicting CKD involves several statistical learning methods, and we have presented their groundwork in detail. We proposed a generalization of Hidden Markov Models, and shown that similar, computationally feasible solutions exist to their classic problems. We have proven the algorithm used in BHMMs to indeed be a monotonically increasing optimization procedure by demonstrating that it is an implementation of Expectation-Maximization.

Using these models, we have developed classifiers for the development of CKD, and shown the lift provided by including information gleaned from diagnosis codes, as opposed to focusing strictly on summary variables. We have demonstrated a basic example of how an insurance company could use these classifiers to make optimal policy choices for increasing profit.

It is our hope that this work will be a useful resource to any student (or faculty member) seeking to better understand Markov chains, convex analysis, Gibbs sampling, Expectation-

Maximization, and Latent Dirichlet Allocation, which is why we have presented these topics so thoroughly. We also hope that our readers will find our generalization of Hidden Markov Models to be approachable and usable in analyzing temporal data.

Modeling and predicting the development of a chronic disease is a nontrivial task, often requiring complex solutions. Our solutions are by no means perfect, and there remains great potential for further improvement. We hope, however, that our work has shown that an approach to these problems can include novel uses or extensions of existing algorithms to areas beyond their original purpose. Including these approaches can improve upon the results that would be obtained by otherwise classical statistical learning procedures.

Bibliography

[1] James Manyika et al. *Big Data: The next frontier for innovation, competition, and productivity.* McKinsey&Company, 2011.

[2] Geoffrey Grimmet and David Stirzaker. *Probability and Random Processes, 3rd Ed.* Oxford University Press, 2001.

[3] Sean Borman. The Expectation Maximization Algorithm: A short tutorial. http://www.seanborman.com/publications/EM_algorithm.pdf, 2009.

[4] Anthony L. Peressini et al. *The Mathematics of Nonlinear Programming.* Springer-Verlag Press, 1988.

[5] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.

[6] A. P. Dempster et al. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 1977.

[7] L. Breiman et al. *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[8] L. Breiman. Random Forests. *Machine Learning*, 2001.

[9] D. Blei et al. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003.

[10] S. Deerwester et al. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 1990.

[11] T. Hofmann. Probabilistic Latent Semantic Indexing. *Proceedings of the Twenty-Second Annual International SIGIR Conference*, 1999.

[12] G. Heinrich. Parameter Estimation for Text Analysis. Version 2.4. Unpublished technical note. http://www.arbylon.net/publications/text-est.pdf, 2008.

[13] B. Carpenter. Integrating Out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling. Version 1.4. Unpublished technical note. http://lingpipe.files.wordpress.com/2010/07/lda3.pdf, 2010.

[14] M. Stamp. A Revealing Introduction to Hidden Markov Models. Unpublished notes. http://www.cs.sjsu.edu/ stamp/RUA/HMM.pdf, 2012.

[15] D. Ramage. Hidden Markov Models Fundamentals. Unpublished notes. http://cs229.stanford.edu/section/cs229-hmm.pdf, 2007.

[16] E.K.P. Chong and S.H. Zak. *An Introduction to Optimization.* John Wiley & Sons, Inc. 2008.