2011-01-01

Bayesian Computational Methods for Hidden Markov Models

Samson Laine Ghebremariam
University of Texas at El Paso, slghebremariam@miners.utep.edu

BAYESIAN COMPUTATIONAL METHODS FOR HIDDEN MARKOV MODELS

Samson L. Ghebremariam

Department of Mathematical Science

APPROVED:

Ori Rosen, Chair, Ph.D.

Ming-Ying Leung, Ph.D.

Max Shpak, Ph.D.

Benjamin C. Flores, Ph.D.
Acting Dean of the Graduate School

BAYESIAN COMPUTATIONAL METHODS FOR HIDDEN MARKOV MODELS

by

Samson L. Ghebremariam

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mathematical Science

THE UNIVERSITY OF TEXAS AT EL PASO

August 2011

Acknowledgements

My deep gratitude goes to my supervisor, Dr. Ori Rosen, who has been an inspiration and instrumental throughout the process. Had his patience, detailed instructions and insightful comments not been included, this thesis would have been impossible. I would also like to thank my thesis committees, Dr. Ming-Ying Leung and Dr. Max Shpak for their time and willingness to be part of my thesis committee.

My sincere thanks goes to many of my fellow graduate students who offered an invaluable help during my study in the program.

Last but not least, I would also like to thank the Mathematical Sciences Department faculty and staff for their support, encouragement and services.

Abstract

Hidden Markov Models (HMMs) have been applied to many real-world problems. Hidden Markov modeling has recently become increasingly important and popular among researchers,and many software tools are based on them. Given that the models are rich in mathematical structure, they can form theoretical foundation for use in a wide range of applications. Hidden Markov models provide a universal configuration for statistical analysis of a large variety of DNA sequences containing symbols A, C, G, T. In a HMM, it is impossible to figure out what state the model is in by just having a look at the symbol generated.

A Bayesian Analysis using Markov chain Monte Carlo (MCMC) sampling techniques can be implemented to simulate the hidden Markov model parameters from their posterior distribution given the observed data. In this paper we adopt a Gibbs sampling algorithm to sample the hidden states and the model parameters from their posterior distribution. Unobservable variables are used as the hidden states indicators. We propose finite mixtures of hidden Markov models with component weights that depend on time. In addition to the hidden state indicators, latent variables are used as mixture indicators. To get a better fit we divide the data into non-overlapping segments of equal lengths such that all the observations within the same segment belong to the same component. The results are compared with the standard MLE estimates through simulation studies and real data and are shown to be better.

Key Words: HMM; Markov switching; EM algorithm; Gibbs sampling; Forward-Backward Algorithms; Metropolis-Hastings; Mixtures-of-experts; Stationary distribution; Periodicity; Reversibility; MCMC; Segmentation; Newton-Raphson.

Table of Contents

List of Figures

List of Tables

Notation

We list the English and Greek letters used in this thesis for reference. These symbols are used in the derivations of the equations in chapters 2, 3 and 4.

Symbol	Notation
$\mathcal{C}$	A set of states.
C_t	The state at time t.
γ_{ij}	The probability of moving from state i to state j.
Γ	The matrix is of a one step transition matrix.
$\Gamma(t)$	The matrix is of a t step transition matrix.
$\gamma_{ij}(t)$	The t-step transition probability from state i to state j.
$\mathbf{u}(t)$	The row vector of probabilities of a Markov chain being in a given state at a particular time t.
$\boldsymbol{\delta}$	The vector representing a stationary distribution.
p_j	The probability of the j^{th} mixture component.
η_j	The parameter of the j^{th} mixture component.
τ_j	Emission parameter for the observation associated with the j^{th} state.
T	Number of observations
m	Number of states
L_T	The likelihood function
y_t	The observation at time t
c_t	The state at time t
$y^{(T)}$	All the observations
$c^{(T)}$	The states under which the observations $y^{(T)}$ are generated
γ_{c_{t-1},c_t}	The probability of transition from a state at time $t-1$ to a state at time t.
δ_{c_1}	The intial probability of the state at time 1.

Symbol	Notation
$\hat{u}_j(t)$	An indicator variable indicating the state at time t.
$\hat{v}_{jk}(t)$	An indicator variable indicating the transition from a state at time $t-1$ to a state at time t.
$\boldsymbol{\alpha}_t$	The vector of forward probabilities at time t.
$\boldsymbol{\beta}_t$	The vector of backward probabilities at time t.
r	The number of possible values generated.
$\boldsymbol{\xi}_j$	The emission probability vectors corresponding to the j^{th} state.
$z_{i,j}$	An indicator variable to indicate that the i^{th} observation is generated from the j^{th} state.
$\mathbf{z}_t$	A vector consisting of indicator variables for the t^{th} observation.
Z	A matrix in which each row contains the indicator variables corresponding to the t^{th} observation.
Θ	An array containing the emission and transition parameters of a HMM.
$\mathbf{e}_j$	An m-vector of all zeros except for 1 in the j^{th} position.
$\text{mult}(1, P)$	A multinomial distribution with 1 as the number of trials and P as the vector of probabilities.
$\mathcal{D}r(a_1, \ldots, a_m)$	A Dirichlet distribution with positive parameters $a_1, a_2, \ldots, a_m$.
$\boldsymbol{\phi}_T$	The ratio $\boldsymbol{\alpha}_T / \boldsymbol{\alpha}_T \mathbf{1}'$.
$\boldsymbol{\varphi_j(h)}$	The j^{th} entry of the vector $\phi_T \Gamma^h$ for $j = 1, \ldots, m$.
h	Forecast horizon.
M	The number of iterations.
B	Burn-in period.
k	The number of mixture components

Symbol	Notation
Θ_j	An array of parameters of the j^{th} mixture component.
$\boldsymbol{\lambda}_j$	A vector of parameters required to specify the weights associated with the j^{th} component of the mixture.
$p_{t,j}$	The probability that the t^{th} observation came from the j^{th} mixture component.
$z_{t,i,j}$	A latent indicator variable to indicate if the t^{th} observation is generated by the i^{th} state from the j^{th} mixture component.
Σ	A hyperparameter for $\boldsymbol{\lambda}_j$
S	The number of segments
$\boldsymbol{\omega}_j$	A vector of parameters required to specify the weights associated with the j^{th} component of the mixture when segmentation is considered.
$p_{s,j}$	The probability that the s^{th} segment belongs to the j^{th} mixture component.
Ω	A matrix made of the vectors $\boldsymbol{\omega}_j$ for all j.
Λ	A matrix made of the vectors $\boldsymbol{\lambda}_j$ for all j.

Chapter 1

Literature Review

This chapter provides an introduction to the Markov processes and hidden Markov models (HMMs). We describe the properties of the Markov processes and the general construction of simple HMMs which will help us as a foundation in our application and modeling with discrete probability HMMs in the subsequent chapters.

1.1 Markov Processes

Stochastic processes are mathematical models that evolve over time in a probabilistic manner. A Markov process is a stochastic process endowed with the Markov property, i.e. the future state only depends on the current state and not on the past history. The process moves from one state to another in a chain-like manner. Markov processes are among the most important of all random processes in which the process can only be in a finite or countable number of states.

We illustrate a Markov process as follows: Given a set of states, $\mathcal{C} = \{ 1, 2, \ldots, m \}$, consider a stochastic process $\{C_t, t = 0, 1, 2, \ldots\}$ that takes values on $\mathcal{C}$. If $C_t = i$, then the process is said to be in state i at time t. If the chain is currently in state i then it moves to state j at the next step with a probability denoted by γ_{ij}, and this probability does not depend upon which states the chain was in before the current state. That is,

$$\gamma_{ij} = P(C_{t+1} = j \mid C_t = i) = P(C_{t+1} = j \mid C_t = i, C_{t-1} = i_{t-1}, \ldots, C_0 = i_0) \tag{1.1}$$

for all i, j, $i_0, \ldots, i_{t-1}$, ϵ $\mathcal{C}$ and t $\geq$ 0. Such a stochastic process is called Markov Process. The states are dependent in a specific way which is mathematically convenient, as

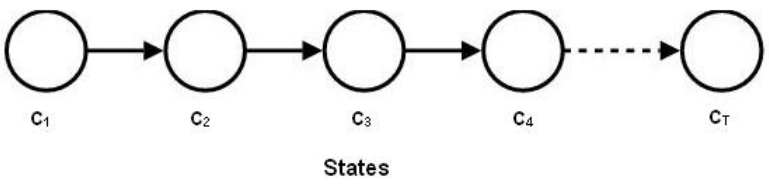

Figure 1.1: Markov chain

shown in Figure 1.1 above, in which the past and the future are dependent only through the present. The probabilities γ_{ij} are known as transition probabilities. The process can remain in the state it is in, and this occurs with probability γ_{ii}. An initial probability distribution $(P(C_0 = 1), \ldots, P(C_0 = m))$ defined on $\mathcal{C}$, specifies the state from which the chain starts. The square matrix of transition probabilities Γ is defined as the matrix with (i, j) element γ_{ij}. The matrix Γ can also be denoted by $\Gamma(1)$ to indicate that the matrix is of a one step transition matrix. It is called a stochastic matrix because the rows sums are equal to 1.
A Markov chain is said to be *homogenous* (Ross, 1996) if the transition probability from time s to time t depends only on the difference $t - s$ for any non-negative integers s and t with $s \leq t$. That is,

$$P(C_t = j \mid C_s = i) = P(C_{t-s} = j \mid C_0 = i), \quad \text{where } i, j \, \epsilon \, \mathcal{C}. \tag{1.2}$$

An important property of all finite state-space homogenous Markov chains is that they satisfy the *Chapman-Kolmogorov equations*

$$\Gamma(t + u) = \Gamma(t)\Gamma(u). \tag{1.3}$$

Proof: The (i,j)th element of the $\Gamma(t+u)$ matrix, $\gamma_{ij}(t+u)$ can be written as

$$\begin{aligned}
\gamma_{ij}(t+u) &= P(C_{t+u} = j \mid C_0 = i) \\
&= \sum_k P\left(C_{t+u} = j, C_u = k \mid C_0 = i\right) \\
&= \sum_k P\left(C_{t+u} = j \mid C_u = k, C_0 = i\right) P\left(C_u = k \mid C_0 = i\right) \\
&= \sum_k P\left(C_{t+u} = j \mid C_u = k\right) P\left(C_u = k \mid C_0 = i\right) \\
&= \sum_k P_{ik}(u) P_{kj}(t) \qquad (1.4)
\end{aligned}$$

where $\Gamma\mathbf{1} = \mathbf{1}$ indicating that the row sums are equal to 1; i.e., the column vector $\mathbf{1}$ is a right eigen vector of Γ and corresponds to an eigenvalue of 1. The number of states of the Markov chain is denoted by m. The row vector of probabilities of a Markov chain being in a given state at a particular time t is denoted by

$$\mathbf{u}(t) = \left(P(C_t = 1), \ldots, P(C_t = m)\right), \quad t \,\epsilon\, \{0, 1, 2, \ldots\}. \qquad (1.5)$$

The vector $\mathbf{u}(0)$ refers to the initial distribution of the Markov chain, that is, at time $t = 0$. The distribution at time $t + 1$ can be deduced from that at time t by the following relation

$$\mathbf{u}(t+1) = \mathbf{u}(t)\Gamma. \qquad (1.6)$$

A Markov chain is said to have a stationary distribution $\boldsymbol{\delta}$ if $\boldsymbol{\delta}\Gamma = \boldsymbol{\delta}$ and $\boldsymbol{\delta}\mathbf{1} = 1$, where the first condition states the stationarity where as the second verifies $\boldsymbol{\delta}$=$(\delta_1, \cdots, \delta_m)$ is a probability distribution (Zucchini and MacDonald, 2009).

Note that the m-step transition probability

$$\gamma_{ij}(m) = P(C_{n+m} = j \mid C_n = i) \quad \text{for some } m, n > 0. \qquad (1.7)$$

State j is said to be accessible from state i if $\gamma_{ij}(m) > 0$ for some $m > 0$. We say that states i and j communicate, denoted by $i \leftrightarrow j$, if states i and j are accessible to each other (*i.e.*, $i \to j$ *and* $j \to i$). States that communicate are classified into the same

communication class. Since communication is an equivalence relation, any two classes are either disjoint or identical.

A Markov chain is said to be irreducible if it consists of only one communication class, that is, if all states communicate with each other.

State i is said to have period d if the following conditions are satisfied

$$\begin{aligned} &(i) \text{ whenever } \gamma_{ij}(m) > 0, \quad m \text{ is a multiple of } d. \\ &(ii) \ d \text{ is the largest integer with the above property.} \end{aligned} \tag{1.8}$$

A state with period 1 is said to be aperiodic. If $i \to j$, then states i and j have the same period, that is, periodicity is a class property (Ross, 1996).

A stochastic process is said to be reversible if its finite dimensional distributions are invariant under reversal of time. A stationary irreducible Markov process with transition matrix Γ and stationary distribution $\boldsymbol{\delta}$ is reversible if and only if (Zucchini and MacDonald, 2009)

$$\delta_i \gamma_{ij} = \delta_j \gamma_{ji} \ \text{ where } i, j \, \epsilon \, \mathcal{C}. \tag{1.9}$$

1.2 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical Markov models in which the distribution that generates an observation depends on an unobserved (hidden) state of a Markov chain (Zucchini and MacDonald, 2009). That is, a HMM is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations (Rabiner, 1989) and (Cappé, Moulines, and Rydén, 2005). HMMs assume that observations are generated from a mixture of distributions among which subjects move according to a latent Markov chain. HMMs are a particular class of dependent mixture models.

The observations in a common finite mixture model are assumed to arise independently from

a distribution with density

$$\psi(y \mid \Theta) = \sum_{j=1}^{m} p_j f_j(y \mid \eta_j), \tag{1.10}$$

where $\sum_{j=1}^{m} p_j = 1$. The $\big(y_i \mid z_i = j\big) \sim f_j(y_i \mid \eta_j)$ are from a parametric family with unknown parameters η_j, $0 < p_j < 1$ and $\Theta = (p_1, \cdots, p_{m-1}, \eta_1, \cdots, \eta_m)$.

HMM is not the only terminology used for this kind of models; (Ephraim and Merhav, 2002) name it'hidden Markov process', (Leroux and Puterman, 1992) call it 'Markov-dependent mixture', and others 'Markov-switching model' or 'Markov mixture model'. In general, the histories of the observations and hidden states from time 1 to time t are represented by $\mathbf{Y^{(t)}}$ and $\mathbf{C^{(t)}}$, respectively. The simplest finite (m-state) HMM model can be summarized by

$$\psi(y_t \mid \tau) = \sum_{j=1}^{m} P(C_t = j \mid C_{t-1}) f(y_t \mid \tau_j). \tag{1.11}$$

where $\tau = (\tau_1, \ldots, \tau_m)$.

The hidden states satisfy

$$P(C_t \mid \mathbf{C}^{(t-1)}) = P(C_t \mid C_{t-1}), \quad t \,\epsilon\, \{1, 2, \ldots\}, \quad \text{(Markov property)},$$

and conditional on the states

$$P(Y_t \mid \mathbf{Y}^{(t-1)}, \mathbf{C}^{(t)}) = P(Y_t \mid C_t), \quad t \,\epsilon\, \{0, 1, \ldots\}.$$

Figure 1.2. shows the general design of an instantiated HMM. The representation consists of two components, namely, 'the state-dependent process' $\{y_t : t = 1, 2, 3 \cdots\}$, and an unobserved 'Markov process' $\{c_t : t = 1, 2, 3 \cdots\}$, where c_t is a homogenous Markov chain on state space $\boldsymbol{c}$. Conditional on c_t, y_t is a series of random variables on $\boldsymbol{y}$ such that the conditional distribution of y_t depends only on c_t, as depicted in Figure 1. Mathematically,

$$y_t \mid c_t \sim f(y \mid \boldsymbol{\eta}_{c_t}). \tag{1.12}$$

In general, the joint distribution of (y_t, c_t), given the past history, can be written as

$$\left(y_t, c_t \mid \mathbf{y}^{(t-1)}, \mathbf{c}^{(t-1)}\right) \sim P(c_t \mid c_{t-1}) f(y_t \mid c_t), \tag{1.13}$$

where $\boldsymbol{y}^{(t-1)} = (y_0, \ldots, y_{t-1})$, $\boldsymbol{c}^{(t-1)} = (c_0, \ldots, c_{t-1})$ and $f(y \mid \boldsymbol{\eta}_{c_t})$ is called the emission probability at time t. (Zucchini and MacDonald, 2009)

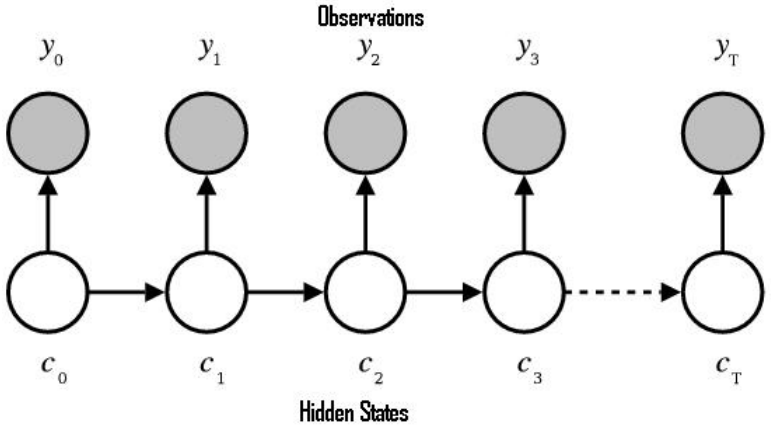

Figure 1.2: Graphical Representation of Hidden Markov Models

1.3 Applications

HMMs have been used in signal processing applications and are especially known for their application in the context of temporal pattern recognition such as speech, handwriting, gesture, face, signature recognition, part-of-speech tagging, telecommunication, partial discharges, wind direction, rainfall, earthquakes, series of daily returns, ion channel modeling and bioinformatics (see e.g., (Geman and Geman, 1983); (Felsenstein and Churchill, 1996); (Juang and Rabiner, 1991); and (He and Kundu, 1991); (Robert, Celeux, and Deibolt, 1993); (Chib, 1996); (Boys and Henderson, 2004)).

HMMs have been used in genetics for modeling DNA sequences for the last few decades. DNA, an acronym to deoxyribonucleic acid, is a nucleic acid that contains the genetic instructions (genes) about a living organism and is replicated in each of its cells. The information in

DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. The order, or sequence, of these bases determines the information available for building and maintaining an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences. In modern molecular biology and genetics, the genome is the entirety of an organism's hereditary information. Life is specified by genomes. It is encoded either in DNA or, for many types of viruses, in RNA. The genome includes both the coding and the non-coding regions of the DNA sequence.

Our example data, Dnadataset is a particular sequence of 9718 chemical bases identical to a complete HIV, or Human Immunodeficiency Virus genome where the bases A, G, C, T are recorded as $1, 2, 3, 4$ (Marin and Robert, 2007). In the Dnadataset, $\mathbf{Y^{(t)}}$ and $\mathbf{C^{(t)}}$ refer to the first t chemical bases generated and the corresponding hidden states under which the chemical bases are generated respectively.

The mixture structure appears because the origin of each observation is unknown. In the particular case, this means that the region to which each chemical base belongs is not identified. The appealing characterstics of HMMs include their simplicity, their general mathematical tractability, and particularly the likelihood is relatively easy to compute.

In this thesis, we develop a model for discrete probability hidden Markov models (HMMs) such as DNA sequences. A Bayesian analysis is implemented using Markov chain Monte Carlo methods (MCMC). We build a mixture model whose components are hidden Markov models with constant but unknown parameters and mixture probabilities that are position-dependent. We assume, without loss of generality, fixed number of components and the same number of states for each component. This model allows parameter shift in DNA sequences where the number of shifts are predetermined.

Several methods have been developed for analyzing hidden Markov models. (Diebolt and Robert, 1993) and (Boys, Henderson, and Wilkinson, 2000) are among those related to our

methodology. (Diebolt and Robert, 1993) developed a method for estimating parameters of a hidden Markov Model. They place non-informative prior distributions on the parameters of HMMs and use Markov chain Monte Carlo simulation to estimate the model for DNA sequences. However, their model does not take into account structural breaks in discrete probability HMMs such as DNA sequences.

Our method differs from the other methods because it allows parameter evolution to handle structural changes in the sequences.

Chapter 2 of this thesis describes the EM algorithm for parameter estimation in HMMs. Chapter 3 presents MCMC methods for parameter estimation and forecasting in a single HMM. Chapter 4 presents in detail the MCMC implementation of the proposed model for piece-wise HMMs and forecasting. Chapter 5 includes simulation results both for single and mixture of HMMs

Our future work will include

- Estimation of the number of states and number of mixture components;
- State prediction;
- Model selection, model checking and outlier detection.

Chapter 2

The EM Algorithm for Parameter Estimation in HMMs

Parameter estimation can be performed by numerical maximization of the likelihood with respect to the parameters. But there are several problems that need to be addressed when the likelihood is computed in this way and maximized numerically in order to estimate the parameters. The main problems are numerical underflow, constraints on the parameters and multiple local maxima in the likelihood function.

2.1 The Estimate-Maximize (EM) Algorithm for HMMs

The EM algorithm (Dempster, Laird, and Rubin, 1977) is a general method for finding the maximum-likelihood estimates of a model where part of the data can be viewed as latent. The EM algorithm applies when the data have missing values or unobserved values or when the likelihood function can be simplified by incorporating latent variables for the missing (or hidden) parameters. The EM algorithm for HMMs makes use of the recursive forward-backward algorithm and is known as Baum-Welch algorithm (Baum, Petrie, Soules, and Weiss, 1970). The likelihood and log likelihood are given, respectively, by, (e.g see (Zucchini

and MacDonald, 2009))

$$L_T = Pr(y^{(T)}, c^{(T)}) = \delta_{c_0} \prod_{t=1}^{T} \gamma_{c_{t-1},c_t} \prod_{t=1}^{T} p_{c_t}(y_t) \tag{2.1}$$

$$\log(Pr(y^{(T)}, c^{(T)})) = \log \left(\delta_{c_0} \prod_{t=1}^{T} \gamma_{c_{t-1},c_t} \prod_{t=1}^{T} p_{c_t}(y_t) \right) \tag{2.2}$$

$$= \log \delta_{c_0} + \sum_{t=2}^{T} \log \gamma_{c_{t-1},c_t} + \sum_{t=1}^{T} \log p_{c_t}(y_t). \tag{2.3}$$

The log-likelihood of the observations $y^{(T)}$ and the missing data $c^{(T)}$ is known as the augmented log-likelihood. The augmented log-likelihood for an HMM factors into three components

$$\begin{aligned} \log(Pr(\boldsymbol{y}^{(T)}, \boldsymbol{c}^{(T)})) &= \sum_{j=1}^{m} u_j(0) \log \delta_j \\ &+ \sum_{j=1}^{m} \sum_{k=1}^{m} \left(\sum_{t=1}^{T} v_{jk}(t) \right) \log \gamma_{jk} \\ &+ \sum_{j=1}^{m} \sum_{t=1}^{T} u_j(t) \log p_j(y_t) \\ &= \text{ component (1) + component (2) + component (3),} \end{aligned} \tag{2.4}$$

where $u_j(t)$ and $v_{jk}(t)$ are indicator variables defined as

$$u_j(t) = \begin{cases} 1 & \text{if } c_t = j \\ 0 & \text{otherwise.} \end{cases}$$

for $t = 1, 2, \ldots, T$,

$$v_{jk}(t) = \begin{cases} 1 & \text{if } c_{t-1} = j \text{ and } c_t = k \\ 0 & \text{otherwise.} \end{cases}$$

for $t = 2, 3, \ldots, T$ and

$\boldsymbol{y}^T$ and $\boldsymbol{c}^T$ represent the observations and hidden states respectively. Components (1), (2) and (3) refer to the initial state distribution, the hidden state transitions, and the emissions, respectively.

Forward and Backward Probabilities

The EM algorithm requires the computation of recursive forward-backward probabilities. The elements of the row vector $\boldsymbol{\alpha}_t$ are known as forward probabilities and are given by (Zucchini and MacDonald, 2009)

$$\boldsymbol{\alpha}_t = \boldsymbol{\delta}\, P(y_1)\, \Gamma P(y_2)\, \Gamma P(y_3)\, \Gamma P(y_4)\, \cdots \Gamma P(y_t) \tag{2.5}$$

for $t = 1, 2, \cdots, T$.

The j^{th} component of $\boldsymbol{\alpha}_t$, the probability of generating the sequence $\boldsymbol{y}^{(t)}$ while ending up in state j, is

$$\alpha_t(j) = P(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3, \cdots, Y_t = y_t, C_t = j) \tag{2.6}$$

for all $j = 1, 2, \cdots, m$.

The vector of backward probabilities $\boldsymbol{\beta}_t$ is defined by

$$\boldsymbol{\beta}_t' = P(y_{t+1})\, \Gamma P(y_{t+2})\, \Gamma P(y_{t+3})\, \Gamma P(y_{t+4}) \cdots \Gamma P(y_T)\, \mathbf{1}' \tag{2.7}$$

for $t = 1, 2, \ldots, T$ and

the j^{th} component of $\boldsymbol{\beta}_t$, the probability of generating the sequence $\boldsymbol{y}_{t+1}^T$, given that at time t, we are in state j is

$$\beta_t(j) = P(Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, Y_{t+3} = y_{t+3}, \cdots, Y_T = y_T \mid C_t = j) \tag{2.8}$$

for all $j = 1, 2, \ldots, m$.

It follows that

$$\alpha_t(j)\beta_t(j) = P(Y_1 = y_1, Y_2 = y_2, \cdots, Y_T = y_T, C_t = j) \text{ and} \tag{2.9}$$

$$L_T = P(\boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) = \boldsymbol{\alpha}_t \boldsymbol{\beta}_t' \tag{2.10}$$

for $t = 1, 2, \ldots, T$ and $i = 1, 2, \ldots, m$.

Since the sequence of states occupied by the Markov-chain component of an HMM is not

observed, a very natural approach to parameter estimation in HMMs is to treat those states as missing data and to employ the EM algorithm in order to find maximum likelihood estimates of the parameters. Indeed, the pioneering work of Leonard Baum and his co-authors (see (Baum et al., 1970); (Baum, 1972); (Welch, 2003)) on what later was called HMMs was an important precursor of the work of (Dempster et al., 1977).
The EM algorithm for HMMs works as follows.

- E step : Compute the conditional expectations of v_{jk}(t) and u_j(t) given the observations $\boldsymbol{y^{(T)}}$ and given the current parameter estimates.

$$\begin{aligned}
\hat{u}_j(t) &= Pr(C_t = j \mid Y^{(T)} = y^{(T)}) \\
&= \frac{Pr(C_t = j, Y^{(T)} = y^{(T)})}{L_T} \\
&= \frac{Pr(C_t = j) Pr(Y_1^t \mid C_t = j) Pr(Y_{t+1}^T \mid C_t = j)}{L_T} \\
&= \frac{Pr(Y_1^t, C_t = j) Pr(Y_{t+1}^T \mid C_t = j)}{L_T} \\
&= \frac{\alpha_t(j)\beta_t(j)}{L_T}
\end{aligned}$$

  and

$$\begin{aligned}
\hat{v}_{jk}(t) &= Pr(C_{t-1} = j, C_t = k \mid Y^{(T)} = y^{(T)}) \\
&= \frac{Pr(C_{t-1} = j, C_t = k, Y^{(T)} = y^{(T)})}{L_T} \\
&= \frac{Pr(Y^{(t-1)}, C_{t-1} = j) Pr(C_t = k \mid C_{t-1} = j) Pr(Y_{t+1}^T \mid C_t = k)}{L_T} \\
&= \frac{\alpha_{t-1}(j)\gamma_{jk} p_k(y_t)\beta_t(k)}{L_T}.
\end{aligned}$$

- M Step: Maximize the augmented likelihood, with respect to the initial distribution $\boldsymbol{\delta}$, the transition probability matrix Γ, and the parameters of state-dependent distributions, with the functions of the missing data in the augmented likelihood replaced by their conditional expectations (as computed in the E-Step).

Maximizing *term 1* with respect to $\boldsymbol{\delta}$

Write $\sum_{j=1}^{m} \hat{u}_j(1) \log \delta_j - \lambda \left(\sum_{j=1}^{m} \delta_j - 1 \right)$ where λ is a Lagrange multiplier. Differentiating each term with respect to δ_j, results in $\hat{u}_j(1)/\delta_j - \lambda = 0$, implying that $\hat{\delta}_j = \hat{u}_j(1) / \sum_{j=1}^{m} \hat{u}_j(1)$.

Maximizing *term 2* with respect to $\boldsymbol{\Gamma}$

Similarly we can use lagrange multipliers to express the constraints $\sum_{j=1}^{m} \gamma_{ij} = 1$ when maximizing $\sum_{j=1}^{m} f_{ij} \log \gamma_{ij}$, where $f_{ij} = \sum_{t=2}^{T} \hat{v}_{ij}(t)$, implying that $\gamma_{ij} = f_{ij} / \sum_{j=1}^{m} f_{ij}$.

Maximizing *term 3* depends on the nature of the state-dependent distributions assumed.

The EM algorithm repeats these two steps until a convergence threshold has been reached.

Chapter 3

Markov Chain Monte Carlo Methods for Parameter Estimation in HMMs

3.1 Introduction

In this chapter we provide a brief introduction to Markov chain Monte Carlo (MCMC) methods for Bayesian parameter estimation in HMMs. In particular, we consider the Gibbs sampler and the Metropolis Hastings (MH) algorithm.

MCMC methods are a class of simulation methods that allow us to simulate a dependent sequence of random draws from probability distributions using Markov chains. Samples of the parameters of even analytically intractable statistical models of interest can be generated (Ryden and Titterington, 1998). MCMC methods provide useful approximations for numerical integration and Bayesian inference. The Gibbs sampler which is very widely applicable to a broad class of Bayesian settings was first introduced by (Geman and Geman, 1984) in the context of image processing and was later discussed in detail by (Tanner and Wong, 1987) in the context of missing-data settings and the data augmentation algorithm. In this chapter, we implement Gibbs sampling in the context of hidden Markov models. Todays Markov chain Monte Carlo (MCMC) techniques allow researchers to implement HMMs without the traditional recursive algorithms, which are viewed as "black boxes by many statisticians (Scott, 2002). Unlike the EM algorithm discussed in Chapter 2, a straightforward Gibbs sampler does not require the tedious forward-backward algorithms.

The Gibbs sampler requires a starting point for the parameters of interest. Given a set of full conditional distributions $\{\pi_1(x_1 \mid x_2, \ldots, x_p), \pi_2(x_2 \mid x_1, x_3 \ldots, x_p), \ldots, \pi_p(x_p \mid x_1, \ldots, x_{p-1})\}$,

the Gibbs sampler simulates successively from these full conditional distributions at each iteration. If the chain is irreducible and aperiodic, it will converge to the stationary distribution $\pi(x_1, \ldots, x_p)$. The properties of the Gibbs Sampler and, in particular, its convergence properties can be checked by using a variety of graphical and numerical diagnostics; see for example, (Cowles and Carlin, 1996), (Brooks, 1998), (Best et al., 1995) and in particular for hidden Markov models (Robert et al., 1993, 1998).

3.2 Estimation via Gibbs Sampling for HMMs

This section presents MCMC methods for sampling the HMM parameters Θ from their posterior distribution given $\boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}$. When both the hidden states, $\mathcal{C} = \{1, \ldots, m\}$ and the observed values, $\mathcal{Y} = \{1, \ldots, r\}$ are finite, as in the case of DNA sequences, let E denote the matrix formed from the m emission probability vectors $\boldsymbol{\xi}_1 = (\xi_{1,1}, \ldots, \xi_{r,1})'$, $\ldots$, $\boldsymbol{\xi}_m = (\xi_{1,m}, \ldots, \xi_{r,m})'$ and as before the parameter matrix Γ is an $m \times m$ matrix of Markov transition probabilities Γ. Thus, $\Theta = (E, \Gamma)$.

3.2.1 The Likelihood Function

We assume, without loss of generality m number of states for the sequence. Also, we assume that the observations are independent and identically distributed (i.i.d) within each state of the hidden chain.

The likelihood function for an m-state HMM is

$$\pi(\Theta \mid y) \propto \sum_{i_1, i_2, i_3, \ldots, i_T = 1}^{m} f(y_1 \mid \boldsymbol{\xi}_{i_1}) \gamma_{i_1, i_2} f(y_2 \mid \boldsymbol{\xi}_{i_2}) \gamma_{i_2, i_3} f(y_3 \mid \boldsymbol{\xi}_{i_3}) \ldots \gamma_{i_{T-1}, i_T} f(y_T \mid \boldsymbol{\xi}_{i_T}), \quad (3.1)$$

where $f(y_t \mid \boldsymbol{\xi}_{i_t})$ is the underlying distribution under which the t^{th} observation is generated and γ_{i_{t-1}, i_t} is the probability of transition from the state at time $t-1$ to the state at time t. The m-tuple sum in the likelihood component involves m^T terms, each of which is a product of $2T$ factors, so is intractable for most values of T.

Each observation y_t, is augmented with a missing value indicator which represents the state

from which y_t is generated. Hence, we define the following indicator variables to indicate the state from which the observation is generated.

$$z_{t,j} = \begin{cases} 1 & \text{if observation } t \text{ came from state } j \quad \text{for } j \in \{1, \ldots, m\}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Let $\mathbf{z}_t = (z_{t,1}, \ldots, z_{t,m})'$, then $Z = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T)$ is assumed to constitute a first-order Markov chain with transition matrix Γ , i.e.

$$P(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \ldots, \mathbf{z}_1) = P(\mathbf{z}_t \mid \mathbf{z}_{t-1}),\ t = 1, \ldots, T.$$

We assume that the incomplete data $\mathbf{Y}^{(T)}$ is observed and is generated by some underlying distribution. We augment Z to $\mathbf{Y}^{(T)}$ and assume that ($\mathbf{Y}^{(T)}$, Z) is a complete dataset. We assume a parametric Model with parameters Θ, and Z specifies the latent or unobserved values in the Model (Mcgrory and Titterington, 2009). We now define a new likelihood function $\pi(\Theta \mid y, Z)$ known as *the augmented likelihood*.

3.2.2 The Augmented Likelihood Function

The augmented likelihood function is

$$\pi(\Theta \mid y, Z) \propto \prod_{i=1}^{T} \prod_{j=1}^{m} \prod_{l=1}^{m} \left(\gamma_{l,j}^{z_{i-1,l}} f(y_i \mid \boldsymbol{\xi}_j) \right)^{z_{i,j}}. \tag{3.3}$$

Note that $z_{0,l} = 0$ for all l.

3.2.3 Prior Specification

Since in our case $f(y_i \mid \boldsymbol{\xi_j})$ is multinomial, a conjugate prior on $\boldsymbol{\xi_j}$ is Dirichlet with parameters $(\alpha_{j,1}, \ldots, \alpha_{j,r})$. Similarly, a conjugate prior on each row of the transition matrix, Γ, is Dirichlet with parameters $(\beta_{j,1}, \ldots, \beta_{j,m})$. Note that r is the maximum value that y_t can take.

3.2.4 Posterior Sampling of Θ=(E, Γ) by MCMC

The augmented posterior density is given as

$$\pi(\Theta \mid \mathbf{y}, Z) \propto \left\{ \prod_{i=1}^{T} \prod_{j=1}^{m} \prod_{l=1}^{m} \left(\gamma_{l,j}^{z_{i-1,l}} f(y_i \mid \boldsymbol{\xi}_j) \right)^{z_{i,j}} \right\} \times \prod_{j=1}^{m} \pi(\boldsymbol{\xi}_j) \pi(\boldsymbol{\gamma}_j). \tag{3.4}$$

We later analyze DNA sequences in which case the distribution of the observed variables given the latent variables

$$\left(y_i \mid \mathbf{z}_i = \mathbf{e}_j\right) \sim \text{mult}(1, \boldsymbol{\xi}_j) \quad \text{for} \quad j \,\epsilon\, \{1, \ldots, m\}, \tag{3.5}$$

where $\mathbf{e}_j$ is an m-vector of all zeros except for 1 in the j^{th} position, mult$(1, P)$ denotes a multinomial distribution with 1 as the number of trials and P as the vector of probabilities. The i^{th} observation, y_i is represented by y_i=$(x_{i,1}, \ldots, x_{i,r})$, that is, $y_i = j$ implies $x_{i,j} = 1$ and $x_{i,l} = 0$ for all $l \neq j$, $j \,\epsilon\, \{1, \ldots, r\}$ and $i \,\epsilon\, \{1, \ldots, T\}$ and $f(y_i = s \mid \xi_j) = \xi_{s,j}$ for $s \,\epsilon\, \{1, \ldots, r\}$.

Then we have,

$$\pi(\Gamma \mid \mathbf{y}, Z) \propto \left\{ \prod_{i=1}^{T} \prod_{j=1}^{m} \prod_{l=1}^{m} \gamma_{l,j}^{z_{i-1,l}\, z_{i,j}} \right\} \times \prod_{j=1}^{m} \pi(\boldsymbol{\gamma}_j) \tag{3.6}$$

$$\pi(E \mid \mathbf{y}, Z) \propto \left\{ \prod_{i=1}^{T} \prod_{j=1}^{m} \left(f(y_i \mid \boldsymbol{\xi}_j) \right)^{z_{i,j}} \right\} \times \prod_{l=1}^{m} \pi(\boldsymbol{\xi}_l), \text{ and} \tag{3.7}$$

$$\pi(Z \mid \mathbf{y}, \Theta) \propto \prod_{i=1}^{T} \prod_{j=1}^{m} \prod_{l=1}^{m} \left(\gamma_{l,j}^{z_{i-1,l}} f(y_i \mid \boldsymbol{\xi}_j) \right)^{z_{i,j}}. \tag{3.8}$$

Thus, the conditional distribution for the l^{th} row of Γ, $\boldsymbol{\gamma}_l = (\gamma_{l,1}, \ldots, \gamma_{l,m})$ is as follows.

$$\pi(\boldsymbol{\gamma}_l \mid y, z) \propto \left\{ \gamma_{l,1}^{\sum_{i=1}^{T} z_{i-1,l} z_{i,1}} \cdots \gamma_{l,m}^{\sum_{i=1}^{T} z_{i-1,l} z_{i,m}} \right\} \times \pi(\boldsymbol{\gamma}_l) \tag{3.9}$$

$$= \left\{ \gamma_{l,1}^{\sum_{i=1}^{T} z_{i-1,l} z_{i,1}} \cdots \gamma_{l,m}^{\sum_{i=1}^{T} z_{i-1,l} z_{i,m}} \right\} \times \prod_{k=1}^{m} \gamma_{l,k}^{\beta l,k} \tag{3.10}$$

$$= \gamma_{l,1}^{\beta_{l,1} + \sum_{i=1}^{T} z_{i-1,l} z_{i,1}} \cdots \gamma_{l,m}^{\beta_{l,m} + \sum_{i=1}^{T} z_{i-1,l} z_{i,m}}, \text{ for } l = 1, \ldots, m. \tag{3.11}$$

Likewise, the conditional distribution of $\boldsymbol{\xi_j}$ is

$$\pi(\boldsymbol{\xi}_j \mid y, z) \propto \left\{ \prod_{i=1}^{T} (f(y_i \mid \boldsymbol{\xi}_j))^{z_{i,j}} \right\} \times \pi(\boldsymbol{\xi}_j) \tag{3.12}$$

$$\propto \prod_{i=1}^{T} (f(y_i \mid \boldsymbol{\xi}_j))^{z_{i,j}} \times \prod_{l=1}^{r} (\xi_{l,j})^{\alpha_{l,j}-1} \tag{3.13}$$

$$\propto \left(\xi_{1,j}\right)^{\alpha_{j,1}+\sum_{i=1}^{T} x_{i,1} z_{i,j}-1} \cdots (\xi_{r,j})^{\alpha_{j,r}+\sum_{i=1}^{T} x_{i,r} z_{i,j}-1} \tag{3.14}$$

Thus, we obtain conditional posterior distributions that are easy to sample from, i.e. since we simulate the initial distribution as given in algorithm (1) and use flat priors on the $\boldsymbol{\xi}_j$ and the $\boldsymbol{\gamma}_j$'s, the posterior distributions are Dirichlet.

$$\boldsymbol{\gamma}_j \sim \mathcal{D}r \left(\beta_{j,1} + \sum_{i=1}^{T} z_{i-1,j} z_{i,1}, \ldots, \beta_{j,m} + \sum_{i=1}^{T} z_{i-1,j} z_{i,m} \right) \quad \text{for } j = 1, \ldots, m \text{ and} \tag{3.15}$$

$$\boldsymbol{\xi}_j \sim \mathcal{D}r \left(\alpha_{j,1} + \sum_{i=1}^{T} x_{i,1} z_{i,j}, \ldots, \alpha_{j,r} + \sum_{i=1}^{T} x_{i,r} z_{i,j} \right) \quad \text{for } j = 1, \ldots, m, \tag{3.16}$$

where $\mathcal{D}r(a_1, \ldots, a_m)$ denotes a Dirichlet distribution with positive parameters $a_1, a_2, \ldots, a_m$. Equation (3.8) is quite convoluted and simulation from it necessitates a time-consuming 'forward-backward' recursion formulae (see (Diebolt and Robert, 1993); (Qian and Titterington, 1991). This difficulty also prohibits the use of *data augmentation* as is in (Tanner and Wong, 1987), i.e. to simulate according to $\pi(Z \mid y, \Theta)$, since

$$\mathbf{P}(\mathbf{z}_T = \mathbf{e}_j \mid \Theta, y, \mathbf{z}_{T-1}, \ldots, \mathbf{z}_1) = \frac{\prod_{l=1}^{m} \gamma_{l,j}^{z_{T-1,l}} f(y_T \mid \boldsymbol{\xi}_j)}{\sum_{i=1}^{m} \prod_{l=1}^{m} \gamma_{l,i}^{z_{T-1,l}} f(y_T \mid \boldsymbol{\xi}_i)}, \tag{3.17}$$

$$\mathbf{P}(\mathbf{z}_{T-1} = \mathbf{e}_j \mid \Theta, y, \mathbf{z}_{T-2}, \ldots, \mathbf{z}_1) = \frac{\prod_{l=1}^{m} \gamma_{l,j}^{z_{T-2,l}} f(y_{T-1} \mid \boldsymbol{\xi}_j) \sum_{i=1}^{m} \gamma_{i,j} f(y_{T-2} \mid \boldsymbol{\xi}_i)}{\sum_{k=1}^{m} \sum_{i=1}^{m} \prod_{l=1}^{m} \gamma_{l,k}^{z_{T-2,l}} \gamma_{i,k} f(y_{T-2} \mid \boldsymbol{\xi}_i) f(y_{T-1} \mid \boldsymbol{\xi}_k)}. \tag{3.18}$$

and complexity grows at each further step. However, the conditional distributions are much

easier to handle, since for $1 < t < T$,

$$\begin{aligned}\mathbf{P}(\mathbf{z}_t = \mathbf{e}_j \mid \Theta, y, \mathbf{z}_{i\neq t}) &= \mathbf{P}(\mathbf{z}_t = \mathbf{e}_j \mid \Theta, y_t, \mathbf{z}_{t-1}, \mathbf{z}_{t+1}) \\ &= \frac{\prod_{i=1}^{m}\prod_{k=1}^{m}\gamma_{i,k}^{z_{t-1,i}\, z_{t,k}} f(y_t \mid \boldsymbol{\xi}_j)\prod_{i=1}^{m}\prod_{k=1}^{m}\gamma_{i,k}^{z_{t,i}\, z_{t+1,k}}}{\sum_{j=1}^{m}\prod_{i=1}^{m}\gamma_{i,j}^{z_{t-1,i}} f(y_t \mid \boldsymbol{\xi}_j)\prod_{l=1}^{m}\gamma_{j,l}^{z_{t+1,l}}}. \end{aligned} \tag{3.19}$$

Thus,

$$\mathbf{P}(\mathbf{z}_t = \mathbf{e}_j \mid \Theta, y_t, \mathbf{z}_{t-1}, \mathbf{z}_{t+1}) \propto \begin{cases} \gamma_{j,j} f(y_1 \mid \boldsymbol{\xi}_j)\prod_{i=1}^{m}\gamma_{j,i}^{z_{2,i}} & \text{if} \quad t = 1 \\ \prod_{i=1}^{m}\gamma_{i,j}^{z_{t-1,i}} f(y_t \mid \boldsymbol{\xi}_j)\prod_{l=1}^{m}\gamma_{j,l}^{z_{t+1,l}} & \text{if} \quad 1 < t < T \\ \prod_{i=1}^{m}\gamma_{i,j}^{z_{T-1,i}} f(y_T \mid \boldsymbol{\xi}_j) & \text{if} \quad t = T \end{cases} \tag{3.21}$$

where $\gamma_{i,j}$ is the probability of transition from the state i to state j. The resulting algorithm is given on the previous page as algorithm 1.

In (2) of the initialization step of algorithm **(Algorithm 1)**, $(\mathbf{z}_t)$ could be randomly generated from $\{1, \ldots, m\}$, however, generating $(\mathbf{z}_t)$ the way it is given in equation (3.21) makes more sense because it is related to the data.

3.3 Forecast Distributions

The forecast distribution of an HMM is the conditional distribution of y_{T+h} given $\mathbf{Y}^{(T)} = \mathbf{y}^{(T)}$ where h is assumed to be the forecast horizon (see Zucchini and MacDonald 2007). In the particular case, of discrete-valued observations, the h-step ahead forecast distribution for the sequence $\mathbf{Y}^T$ is computed as a ratio of likelihoods (see Zucchini and MacDonald, 2007) and is given by

$$Pr(y_{T+h} = y \mid \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) = \frac{Pr(y_{T+h} = y, \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)})}{Pr(\boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)})} \tag{3.22}$$

$$= \frac{\boldsymbol{\delta} P(y_1)\Gamma P(y_2)\cdots\Gamma P(y_T)\Gamma^h P(y)\mathbf{1}'}{\boldsymbol{\delta} P(y_1)\Gamma P(y_2)\cdots\Gamma P(y_T)\mathbf{1}'} \tag{3.23}$$

$$= \frac{\boldsymbol{\alpha}_T \Gamma^h P(y)\mathbf{1}'}{\boldsymbol{\alpha}_T \mathbf{1}'}. \tag{3.24}$$

Algorithm 1 : Gibbs Sampling for A Finite State HMM

Initialization :

1. Initialize Θ^0.

2. Initialize Z by generating the hidden Markov chain $(z_t)_{0\leq t<T}$ from $(j = 1, \ldots, m)$

$$\mathbf{P}(\mathbf{z}_t = \mathbf{e}_j \mid \Theta^1, y_t) \propto \begin{cases} \gamma_{j,j} f(y_0 \mid \boldsymbol{\xi}_j) & \text{if } t = 0 \\ \prod_{i=1}^m \gamma_{i,j}^{z_{t-1,i}} f(y_t \mid \boldsymbol{\xi}_j) & \text{if } t > 0 \end{cases} \tag{3.20}$$

and compute the corresponding statistics as

$$\sum_{i=1}^T x_{i,l}\{z_{ij}\}, \quad \sum_{i=2}^T z_{i-1,j}\, z_{i,k} \quad j, k \,\epsilon\, \{1, \ldots, m\} \text{ and } \; l \,\epsilon\, \{1, \ldots, r\}$$

Iteration m: $m > 1$

1. Generate

$$\gamma_j \propto \mathcal{D}r \left(\beta_{1,j} + \sum_{i=1}^T z_{i-1,j} z_{i,1}, \ldots, \beta_{m,j} + \sum_{i=1}^T z_{i-1,j} z_{i,m} \right) \text{ for } j = 1, \ldots, m.$$

$$\boldsymbol{\xi}_j \propto \mathcal{D}r \left(\alpha_{1,j} + \sum_{i=1}^T x_{i,1} z_{i,j}, \ldots, \alpha_{r,j} + \sum_{i=1}^T x_{i,r} z_{i,j} \right) \quad \text{for} \quad j = 1, \ldots, m.$$

and correct the missing initial probability by a Metropolis-Hastings step.

2. Generate successively $\underline{z}$ by generating the hidden Markov chain $(z_t)_{0\leq t<T}$ from $(j = 1, \ldots, m)$

$$\mathbf{P}(\mathbf{z}_t = \mathbf{e}_j \mid \Theta, y_t, \mathbf{z}_{t-1}, \mathbf{z}_{t+1}) \propto \begin{cases} \gamma_{j,j} f(y_1 \mid \boldsymbol{\xi}_j) \prod_{i=1}^m \gamma_{j,i}^{z_{1,i}} & \text{if} \quad t = 1 \\ \prod_{i=1}^m \gamma_{i,j}^{z_{t-1,i}} f(y_t \mid \boldsymbol{\xi}_j) \prod_{l=1}^m \gamma_{j,l}^{z_{t+1,l}} & \text{if} \quad t > 1 \\ \prod_{i=1}^m \gamma_{i,j}^{z_{T-1,i}} f(y_T \mid \boldsymbol{\xi}_j) & \text{if} \quad t = T \end{cases}$$

and compute the corresponding statistics as in (2) of the initialization step.

where $P(y_t)$ is a diagonal matrix of the state-dependent distributions of y_t.
Letting $\boldsymbol{\phi}_T = \boldsymbol{\alpha}_T / \ \boldsymbol{\alpha}_T \mathbf{1}'$ where $\mathbf{1} = (1, \ldots, 1)$, we have

$$Pr(y_{T+h} = y \mid \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) = \phi_T \Gamma^h P(y) \mathbf{1}'. \tag{3.25}$$

The forecast distribution can therefore be written as a mixture of the state-dependent probability distributions

$$Pr(y_{T+h} = y \mid \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) = \sum_{j=1}^{m} \boldsymbol{\varphi}_j(h) p_j(y), \tag{3.26}$$

where the state-dependent distribution

$$p_j(y) = Pr(y_{T+h} = y \mid \mathbf{z}_{T+h} = \mathbf{e}_j), \text{ and}$$

the weight $\boldsymbol{\varphi}_j(h)$ is the j^{th} entry of the vector $\phi_T \Gamma^h$ for $j = 1, \ldots, m$. That is, $p_j(y)$ is the probability mass function of y_{T+h} given that the Markov chain is in state j at time $T + h$. We make use of forward probabilities in calculating the forecast distribution of the observations. To estimate the h-step forecast distribution of the sequence, we compute the distribution at each Gibbs sampling iteration and take the average over the iterations after a specified burn-in period, B (usually 2000 in $10,000$ iterations). That is, the h-step forecast distribution of the sequence is given by

$$\hat{Pr}(y_{T+h} = y \mid \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) = \frac{1}{M - B - 1} \sum_{i=B+1}^{M} \sum_{j=1}^{m} \boldsymbol{\varphi}_j^l(h) p_j^l(y). \tag{3.27}$$

where M is large enough number of iterations and B is the burn-in period. We do this both for the real data and simulated ones.

Chapter 4

Mixtures of Hidden Markov Models

4.1 Introduction

In this chapter, we are concerned with modeling and making inference for HMMs with structural breaks. We propose a finite mixture model for analyzing HMMs whose parameters change over the sequence. A Bayesian method for fitting the model is implemented using MCMC techniques. We assume, for simplicity that the number of components in the mixture are known but the parameters of the hidden Markov models are unknown. Our goal is to improve forecast distributions and state prediction using mixture modeling, compared to forecast distributions and state prediction based on a model with a single component. We introduce a new method using mixtures-of-experts models (Jacobs, Jordan, Nowlan, and Hinton, 1991), i.e., mixture models in which both mixture component probabilities and mixture components have covariates. The sequence is divided into non-overlapping segments to improve identification of changes in the sequence.

In addition to the unobservable indicators described in Chapter 3, the data will further be augmented with mixture indicators.

4.1.1 Mixture of HMMs

We propose the k-component mixture model

$$f(y_t \mid \Theta, \Lambda) = \sum_{j=1}^{k} p_{t,j} h_j(y_t \mid \Theta_j), \tag{4.1}$$

where $\sum_{j=1}^{m} p_{t,j} = 1$, $0 < p_{t,j} < 1$, $m > 1$, $f(\cdot)$ is a pdf/pmf, the h_j are pdfs/pmfs of hidden Markov models with unknown parameter Θ_j. The array (Θ, Λ) contains parameters of the

mixture model where $\Theta = (\Theta_1, \ldots, \Theta_k)$, Θ_j is the parameter array for the j^{th} component of the mixture, and $\Lambda = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k)$, $\boldsymbol{\lambda}_j$ is a vector of parameters required to specify the weights associated with the j^{th} component of the mixture. The corresponding likelihood function is

$$\prod_{t=1}^{T} \sum_{j=1}^{k} p_{t,j} h_j(y_t \mid \Theta_j). \tag{4.2}$$

This model belongs to the class of mixtures-of-experts models (Jacobs et al., 1991). The mixtures-of-experts models assume finite mixture distribution (Peng, Jacobs, and Tanner, 1996).The $p_{t,j}$ are the weights associated with the j^{th} component at time t in a mixture of k hidden Markov models. See Section 4.3 for more details on these weights.

4.2 Model and Priors for Hidden Markov Components

4.2.1 Model

The model is given in equation 4.1 where we assume, without loss of generality, the same number, m, of states for each component. The j^{th} component of the mixture model is a HMM with parameter array Θ_j where $\Theta_j = (\Gamma_j, E_j)$. The matrix of transition parameters for the j^{th} hidden Markov model is denoted by Γ_j and the matrix of emission parameters E_j is formed from the m probability vectors $\boldsymbol{\xi}_{1,j} = (\xi_{1,j,1}, \ldots, \xi_{r,j,1})$, $\ldots$, $\boldsymbol{\xi}_{m,j} = (\xi_{1,j,m}, \ldots, \xi_{r,j,m})$. The $(i,l)^{th}$ entry of the matrix Γ_j is denoted by $\gamma_{i,l,j}$ and $\boldsymbol{\xi}_{i,j}$ corresponds to the i^{th} hidden state of the j^{th} components of the mixture.

4.2.2 Priors on $\Theta_j = (\Gamma_j, E_j)$

The same priors from equation (3.2.3) are used here. We consider non-informative priors so we set both $a_{1,j,i} = \cdots = a_{r,j,i} = 1$ and $b_{i,1,j} = \cdots = b_{i,m,j} = 1$.

4.3 Model and Priors for the Mixture Weights

The mixing weights depend on time and on the unknown parameter matrix $\underline{\Lambda} = (\boldsymbol{\lambda}_1', \ldots, \boldsymbol{\lambda}_k')$ through the multinomial logit function (Wood, Rosen, and Kohn, 2011)

$$p_{t,j} = \frac{\exp(\boldsymbol{\lambda}_j' \boldsymbol{t})}{\sum_{h=1}^{k} \exp(\boldsymbol{\lambda}_h' \boldsymbol{t})}, \tag{4.3}$$

where $\boldsymbol{t} = (1, \frac{t}{T})$ and k is the number of components of the mixture. We set $\boldsymbol{\lambda}_k = \mathbf{0}$ for identifiability.

4.3.1 Priors on Λ

The priors on $\boldsymbol{\lambda}_j$, $j = 1, \ldots, k-1$, are independent bivariate normal $N(0, \Sigma)$, where $\Sigma = \sigma^2 I_2$ and $\sigma^2 = 10$.

4.4 Posterior Sampling of the Mixture Model Parameters by MCMC

All the parameters of the k-component mixture model are sampled from their posterior distribution. We introduce latent indicator variables to indicate the component to which an observation belongs.

$$q_{t,j} = \begin{cases} 1 & \text{if } y_t \text{ is generated by the } j^{th} \text{ mixture component} \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

The augmented likelihood is

$$L(\Theta, \lambda \mid y, z) = \prod_{t=1}^{T} \prod_{j=1}^{k} \left\{ p_{t,j} h_j(y_t \mid \Theta_j) \right\}^{q_{t,j}}. \tag{4.5}$$

Thus, the augmented posterior density is given as

$$\pi(\Theta, \lambda \mid y, z) = \prod_{t=1}^{T}\prod_{j=1}^{k}\left\{p_{t,j}h_j(y_t \mid \Theta_j)\right\}^{q_{t,j}} \tag{4.6}$$

$$\times \prod_{j=1}^{k}\prod_{i=1}^{m}\pi(\boldsymbol{\xi}_{ij})\left(\prod_{l=1}^{m}\pi(\boldsymbol{\gamma}_{lj})\right)\cdot\pi(\boldsymbol{\lambda}'_j). \tag{4.7}$$

where $\boldsymbol{\xi}_{ij}$ is the i^{th} row of E_j and $\boldsymbol{\gamma}_{lj}$ is the l^{th} row of Γ_j. Then we have,

$$\begin{aligned}\pi(\{\boldsymbol{\lambda}'_j\}_{j=1,\ldots,k} \mid y, z) &\propto \prod_{t=1}^{T}\prod_{j=1}^{k}\left\{p_{t,j}^{q_{t,j}} \times \pi(\boldsymbol{\lambda}'_j)\right\},\\ &= \prod_{t=1}^{T}\prod_{j=1}^{k}\left(\frac{\exp(\boldsymbol{\lambda}'_j\boldsymbol{t})}{\sum_{h=1}^{k}\exp(\boldsymbol{\lambda}'_h\boldsymbol{t})}\right)^{q_{t,j}}\\ &\times \frac{1}{2\pi\mid\Sigma\mid^{1/2}}\exp\left\{-\frac{1}{2}\boldsymbol{\lambda}'_j\Sigma^{-1}\boldsymbol{\lambda}_j\right\},\end{aligned}$$

where $\mid\Sigma\mid$ is the determinant of Σ and (4.8)

$$\begin{aligned}\pi(\{\Gamma_j, E_j\}_{j=1,\ldots,k} \mid y, z) &\propto \prod_{t=1}^{T}\prod_{j=1}^{k}h_j(y_t \mid \Gamma_j, E_j)^{q_{t,j}}\\ &\times \prod_{i=1}^{m}\pi(\boldsymbol{\xi}_{ij})\left(\prod_{l=1}^{m}\pi(\boldsymbol{\gamma}_{lj})\right),\end{aligned}$$

where $E_j = (\boldsymbol{\xi}_{1j}, \ldots, \boldsymbol{\xi}_{mj})$ and $\Gamma_j = (\boldsymbol{\gamma}'_{1j}, \ldots, \boldsymbol{\gamma}'_{mj})$.
We simulate the j^{th} row of Γ_l and the j^{th} vector of E_l as given in equations (3.15) and (3.16) respectively. However, equation (4.8) is analytically intractable so we implement a Metropolis Hastings step (Wood et al., 2011).

4.4.1 Metropolis Hastings Step

We use Matlab's mnrfit function and optimize equation (4.8) neglecting the prior part to obtain the matrix of coefficient estimates $\hat{\Lambda}$ for the multinomial logistic regression. We also obtain the estimated variance-covariance matrix $\hat{\Sigma}$ for this matrix of coefficients by computing the inverse of the observed information matrix. The Metropolis Hastings step is

then performed as follows

Starting with $\underline{\Lambda}^{(0)} = (\boldsymbol{\lambda}_1'^{(0)}, \ldots, \boldsymbol{\lambda}_k'^{(0)})$ at iteration t of the Gibbs sampler

1. Draw $\Lambda^{(prop)} \sim q(\cdot \mid \Lambda^{(t-1)})$ where $q(\cdot \mid \Lambda^{(t-1)})$ is a multivariate normal distribution $N(\hat{\Lambda}, \hat{\Sigma})$ and the superscript $(prop)$ on a Λ denotes a newly proposed value for Λ.

2. Compute

$$\rho(\Lambda^{(prop)} \mid \Lambda^{(t-1)}) = min \left\{ 1, \frac{\pi(\Lambda^{(prop)}) \cdot q(\Lambda^{(t-1)} \mid \Lambda^{(prop)})}{\pi(\Lambda^{(t-1)}) \cdot q(\Lambda^{(prop)} \mid \Lambda^{(t-1)})} \right\},$$

   where $\pi(\Lambda^{(prop)})$ and $\pi(\Lambda^{(t-1)})$ denote the likelihood of Λ evaluated at the proposal and current values respectively, and $q(\Lambda^{(prop)} \mid \Lambda^{(t-1)})$ and $q(\Lambda^{(t-1)} \mid \Lambda^{(prop)})$ refer to the proposal density evaluated at the proposed and current values respectively.

3. With probability $\rho(\Lambda^{(prop)} \mid \Lambda^{(t-1)})$ set $\Lambda^{(t)} = \Lambda^{(prop)}$, otherwise set $\Lambda^{(t)} = \Lambda^{(t-1)}$.

An alternative to the mnrfit function is a Newton-Raphson Method.

4.5 Data Segmentation

Data segmentation means partitioning the data into segments possibly of different lengths. In i.i.d. mixture models, the latent allocation of observations to mixture components is made by computing the probabilities that the observations came from the components. Unlike in i.i.d. mixture models, HMMs do not assume independence. Thus, in mixture of MMs the latent allocation of observations to the components must take into account this dependence, and for this reason, we follow (Wood et al., 2011) in partitioning the sequence into S small segments, each of length L. In particular, all observations y_t for $t \, \epsilon \, \{1 + (s-1)L, \ldots, sL\}$ belong to the same segment s, $s = 1, \ldots, S$, and allocation of each segment to a j^{th} component of the mixture is accomplished by computing the probability that the segment was generated from the j^{th} HMM. Thus, the general mixture model in equation (4.1) becomes

$$f(y_t \mid (\Theta, \underline{\mathrm{W}})) = \sum_{j=1}^{k} p_{s,j} h_j(y_t \mid \Theta_j), \tag{4.9}$$

for all $t\,\epsilon\,\{1 + (s-1)L, \ldots, sL\}$ and for all $s\,\epsilon\,\{1, \ldots, S\}$, $S = \frac{T}{L}$. Unlike in equation (4.1) where the mixing weights are functions of time, here they are functions of the segments s, $s\,\epsilon\,\{1, \ldots, S\}$. Again, the mixing weights depend on the unknown parameter matrix $\underline{\mathrm{W}} = (\boldsymbol{\omega}_1', \ldots, \boldsymbol{w}_k')$ and on the segment and have multinomial logit form. Hence, equation (4.3) becomes

$$p_{s,j} = \frac{\exp(\boldsymbol{w}_j' \boldsymbol{s})}{\sum_{h=1}^{k} \exp(\boldsymbol{w}_h' \boldsymbol{s})} \tag{4.10}$$

where $\boldsymbol{s} = (1, \frac{s}{S})$.

The corresponding likelihood function in equation (4.2) becomes

$$\prod_{s=1}^{S} \prod_{t=1+(s-1)L}^{sL} \sum_{j=1}^{k} p_{s,j} h_j(y_t \mid \Theta_j.) \tag{4.11}$$

(Wood et al., 2011) recommend that in selecting the segment length, L, it is necessary that L satisfy the following two criteria

(i) L contains enough observations to estimate the dependence in the sequence and

(ii) L is as small as possible to accurately detect changes in the sequence.

4.6 The Sampling scheme

We introduce latent indicator variables to specify the component from which a segment is generated. That is,

$$q_{s,j} = \begin{cases} 1 & \text{if } y_t,\ t = 1 + (s-1)L, \ldots, sL \text{ is generated by the } j^{th} \text{ mixture component} \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Note that $q_{s,j} = 1$ implies $q_{t,j} = 1$ for all $t \, \epsilon\{1 + (s-1)L, \ldots, sL\}$.

Now, let $\mathbf{q}_s = (q_{s,1}, \ldots, q_{s,k})'$ for $j = 1, \ldots, m$, then $Q = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_S)$. The augmented likelihood is

$$L(\Theta, \underline{\mathrm{W}} \mid \mathbf{y}, Q) = \prod_{s=1}^{S} \prod_{j=1}^{k} \left\{ p_{s,j} \prod_{t=1+(s-1)L}^{sL} h_j(y_t \mid \Theta_j) \right\}^{q_{s,j}}. \tag{4.13}$$

Hence, the augmented posterior density is given as

$$\begin{aligned} \pi(\Theta, \underline{\mathrm{W}} \mid \mathbf{y}, Q) = & \prod_{s=1}^{S} \prod_{j=1}^{k} \left\{ p_{s,j} \prod_{t=1+(s-1)L}^{sL} h_j(y_t \mid \Theta_j) \right\}^{q_{s,j}} \\ & \times \prod_{j=1}^{k} \prod_{i=1}^{m} \pi(\xi_{i,j}) \left(\prod_{l=1}^{m} \pi(\boldsymbol{\gamma}_{i,l,j}) \right) \cdot \pi(\boldsymbol{w}_j). \end{aligned} \tag{4.14}$$

We sample all the parameters of our mixtures-of-experts model from their posterior distribution.

4.6.1 Sampling Algorithm

1. Initialization Step:

   - Set k and initialize Q.

2. Iteration Step: iterate the following draws M number of times.

   - Draw the j^{th} row of Γ_l from the Dirichlet distribution $\pi(\gamma_{j,l} \mid \mathbf{y}, \mathbf{q}_{.l})$, for $j = 1, \ldots, m$ and $l = 1, \ldots, k$ where $\mathbf{q}_{.l}$ refers to all the segments that belong to the l^{th} mixture component.

   - Draw the j^{th} column of E_l from the Dirichlet distribution $\pi(\xi_{j,l} \mid \mathbf{y}, \mathbf{q}_{.l})$, for $j = 1, \ldots, m$ and $l = 1, \ldots, k$.

   - Draw $\boldsymbol{w}_j$ from multivariate normal distribution $\pi(\boldsymbol{w}_j, \mid Q)$, for $j = 1, \ldots, m-1$.

   - Draw $\mathbf{q}_s$ from the multinomial distribution $\pi(\mathbf{q}_s \mid \Theta, \underline{\mathrm{W}})$, for $s = 1, \ldots, S$.

4.7 Forecast Distributions for the Mixture Model

We compute the forecast distribution of the k-component mixture model at each iteration of the Gibbs Sampling as

$$Pr(y_{T+h} = y \mid Y^{(T)} = y^{(T)}) = \sum_{j=1}^{k} p_{S+n,j} f_j(y_{T+h} = y \mid Y^{(T)} = y^{(T)}, \Gamma_j, E_j), \quad h \geq 1, \quad (4.15)$$

where $n = \lceil \frac{h}{L} \rceil$, $p_{S+n,j}$ is caculated as given in equation (4.10) with s^* replaced by $S + n^*$ and f_j is the contribution of the j^{th} component to the forecast distribution of the mixture model. The forecast distribution of the j^{th} component of the mixture, f_j is calculated as given in equation (3.26). Thus, we take the average of the forecast distribution over the iterations for some specified burn-in period, usually taken to be 2,001 in 10,000 iterations. That is, the h-step forecast distribution of the sequence based on the mixture model is given by

$$\hat{Pr}(y_{T+h} = y \mid \mathbf{Y^{(T)}} = \mathbf{y^{(T)}}) = \frac{1}{M - B - 1} \times \sum_{l=B+1}^{M} \sum_{j=1}^{k} p_{S+n,j} f_j(y_{T+h} = y \mid Y^{(T)} = y^{(T)}, \Gamma_j^{(l)}, E_j^{(l)}), \quad (4.16)$$

where M is large enough number of iterations and B is the burn-in period, again $p_{S+n,j}$ is caculated as given in equation (4.10) with $\underline{\mathrm{W}}$ replaced by $\underline{\mathrm{W}}^{(l)}$. The ($\Gamma^{(l)}$, $E^{(l)}$, $\underline{\mathrm{W}}^{(l)}$) are the MCMC iterates.

Chapter 5

Simulation

5.0.1 Simulation of Model Parameters for a Single HMM

For Dnadataset, we assume a two-state hidden Markov model, one state corresponding to the coding regions and the other to the non-coding regions of the DNA sequence. Also, we have four possible values (1, 2, 3, 4) generated, each corresponding to the four chemical bases: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) respectively. That is, in this particular case, we have $m = 2$ and $r = 4$. In Table 5.1 and Table 5.2, $\hat{\gamma}_{i,i}$ indicates the estimated transition probability from state i to itself and $\hat{\xi}_{l,i}$ denotes the estimated probability of emitting the symbol l in state i for all i and l.

Table 5.1: Posterior means of the parameters for the Dnadataset sequence modeled as a single HMM along with the log-likelihood based on 10,000 iterations of the Gibbs sampler. The values enclosed in parenthesis in each row are the starting values corresponding to the estimates in the line above them.

$Runs$	$\hat{\gamma}_{1,1}$	$\hat{\gamma}_{2,2}$	$\hat{\xi}_{1,1}$	$\hat{\xi}_{2,1}$	$\hat{\xi}_{3,1}$	$\hat{\xi}_{1,2}$	$\hat{\xi}_{2,2}$	$\hat{\xi}_{3,2}$	Log-likelihood
1	0.688	0.589	0.379	0.017	0.411	0.313	0.405	0.016	-13,117
	(0.985)	(0.796)	(0.362)	(0.182)	(0.250)	(0.239)	(0.261)	(0.189)	
2	0.680	0.595	0.380	0.013	0.423	0.313	0.399	0.018	-13,117
	(0.860)	(0.373)	(0.276)	(0.205)	(0.272)	(0.650)	(0.082)	(0.110)	
3	0.687	0.593	0.373	0.014	0.420	0.321	0.402	0.015	-13,117
	(0.239)	(0.093)	(0.051)	(0.187)	(0.702)	(0.055)	(0.437)	(0.145)	
4	0.672	0.610	0.377	0.016	0.430	0.317	0.379	0.027	-13,118
	(0.722)	(0.265)	(0.037)	(0.263)	(0.623)	(0.536)	(0.072)	(0.001)	
5	0.702	0.584	0.377	0.025	0.411	0.314	0.407	0.010	-13,118
	(0.980)	(o.77)	(0.350)	(0.047)	(0.157)	(0.301)	(0.205)	(0.420)	

In our application, a chain of a parameter started at random does converge to the same values as the Gibbs chains simulated, as shown by Table 5.1 and Figure 5.1 below. For most starting values, convergence is reached based on 1,000 iterations. Nonetheless, for some starting values, large number of iterations are needed to achieve complete stability. Our results for the Dnadataset sequence agree with the results obtained by (Marin and Robert, 2007).

We also simulate a sequence with a given transition and emission probabilities. Table 5.2 shows the maximum likelihood (MLE) and Bayes estimates for the simulated sequence. The mean squared error (MSE) of both estimates are computed in three runs using different starting values in each run and are given in the rightmost column of table 5.2. The considerably smaller MSEs of the Bayes estimates indicate that the Bayes estimates are better than the MLEs. For comparison purposes, the MLEs and Bayes estimates are computed based on the same number of iterations and same starting values in each run.

Table 5.2: Three runs for computing the MLEs and Bayes estimates of the parameters for the simulated sequence modeled as a single HMM, along with their corresponding MSEs and true values based on 1,000 iterations. The values enclosed in parenthesis in each row are the MLEs corresponding to the Bayes estimates on the line above them.

Runs	$\hat{\gamma}_{1,1}$	$\hat{\gamma}_{2,2}$	$\hat{\xi}_{1,1}$	$\hat{\xi}_{1,2}$	$\hat{\xi}_{1,3}$	$\hat{\xi}_{2,1}$	$\hat{\xi}_{2,2}$	$\hat{\xi}_{2,3}$	MSE
1	0.816	0.901	0.266	0.217	0.126	0.277	0.232	0.087	2.85e-03
	(0.906)	(0.996)	(0.172)	(0.301)	(0.002)	(0.276)	(0.221)	(0.110)	8.94e-03
2	0.802	0.903	0.297	0.167	0.151	0.256	0.260	0.0772	2.57e-03
	(0.608)	(0.907)	(0.360)	(0.138)	(0.173)	(0.252)	(0.245)	(0.091)	1.10e-02
3	0.935	0.896	0.284	0.203	0.131	0.248	0.265	0.0552	3.19e-03
	(0.897)	(0.994)	(0.182)	(0.327)	(0.007)	(0.277)	(0.219)	(0.112)	9.31e-03
True Values	0.89	0.99	0.3333	0.1667	0.1667	0.27	0.23	0.1	

We compute the forecast distribution for Dnadataset and the simulated sequences. As the forecast horizon h increases, the forecast distribution converges to the marginal distribution

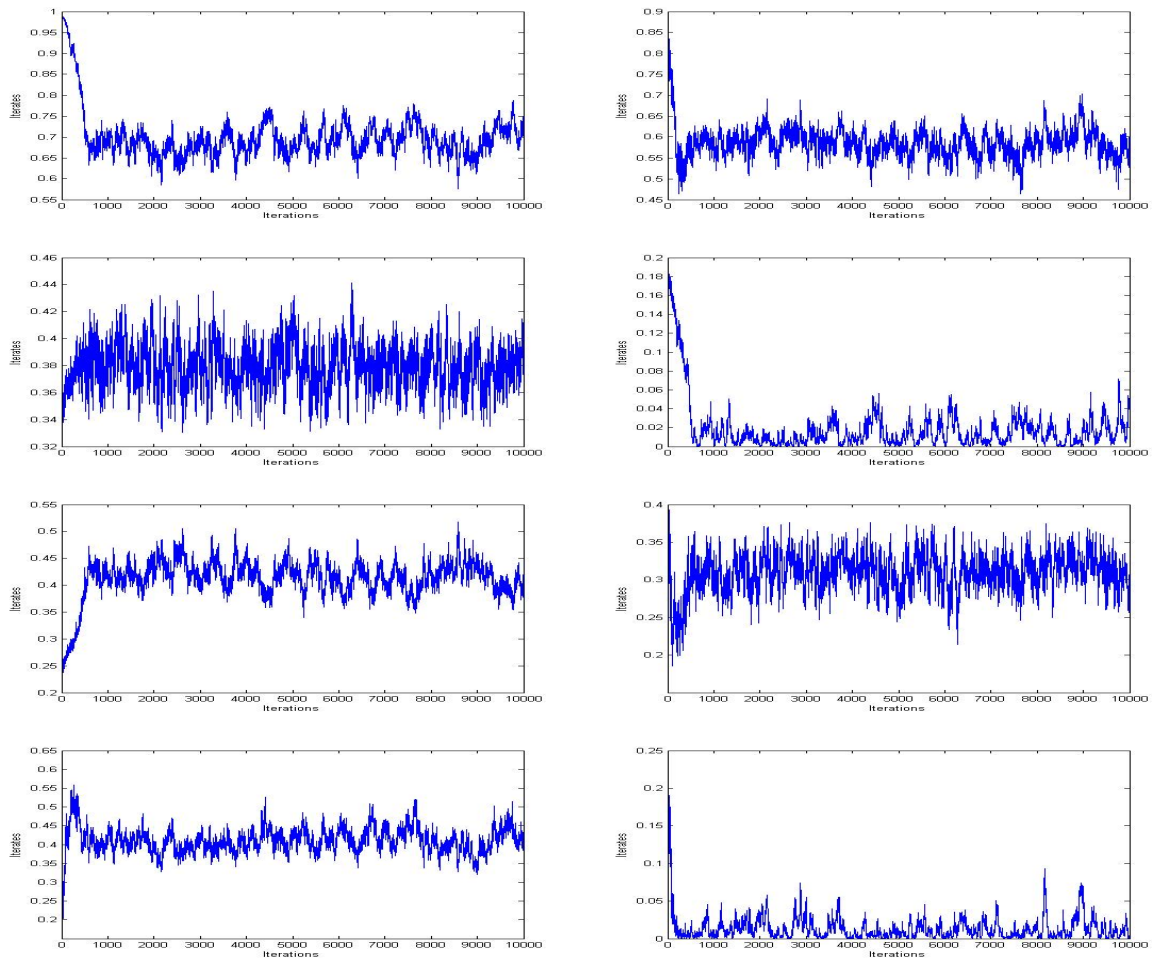

Figure 5.1: Gibbs sampler iterates for the first run for Dnadataset sequence that results in the Bayes posterior means given in the first row of Table 5.1. The figures above show convergence of each of the Gibbs sampler iterates over the iterations. The figures are ordered row-wise to match the order of the parameters in the first row of Table 5.1.

of the stationary HMM (Zucchini and MacDonald, 2009), i.e.,

$$\begin{aligned}\lim_{h\to\infty} Pr(y_{T+h} = y \mid \boldsymbol{Y}^{(T)} = \boldsymbol{y}^{(T)}) &= \lim_{h\to\infty} \phi_T \Gamma^h P(y)\mathbf{1}' \\ &= \boldsymbol{\delta}^* P(y)\mathbf{1}', \end{aligned} \tag{5.1}$$

where we use $\boldsymbol{\delta}^*$ to represent the stationary distribution of the Markov chain. The limit follows from the observation that for any nonnegative row vector $\mathbf{b}$ whose entries add to

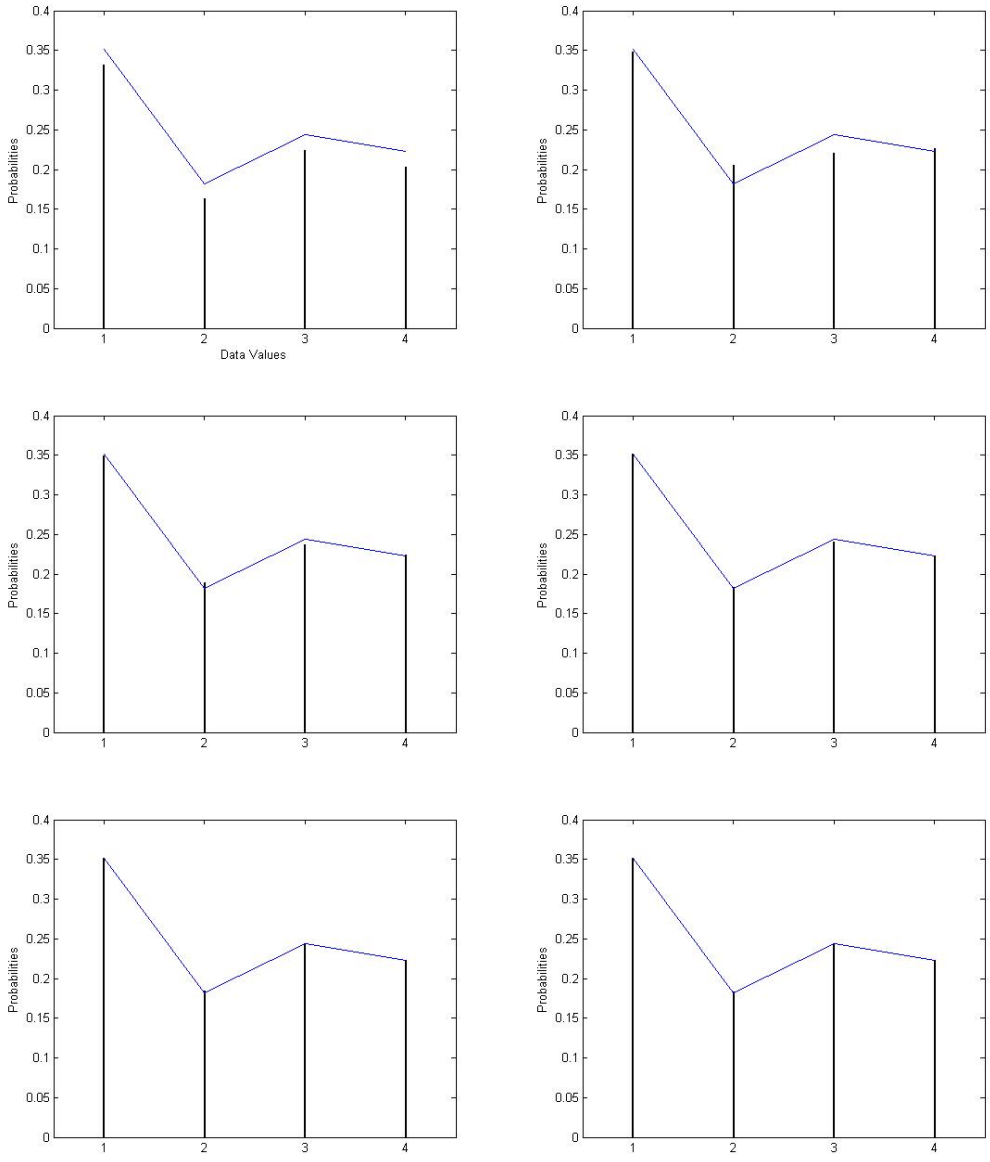

Figure 5.2: Dnadataset, *two*-state multinomial-HMM: forecast distributions for forecast horizons h =1, 2, 3, 4, 5, 6 ahead, compared to limiting distribution, which is shown as a continous line. The figures are ordered row-wise to match the order of forecast horizons.

1, the vector $\mathbf{b}\Gamma^h$ approaches the stationary distribution of the Markov chain as $h \to \infty$, provided the Markov chain satisfies the usual regularity conditions of irreducibility and aperiodicity.

In our application, the forecast distribution rapidly approaches its limiting distribution. Figure 5.2 shows six of the forecast distributions for Dnadataset, compared with the limiting distribution.

5.0.2 Simulated sequences for a mixture of HMMs

A sequence of length 20,000 is generated as

$$\underline{\mathbf{y}} = \begin{cases} HMM1 & \text{for } t = 1, \ldots, 10^4, \\ HMM2 & \text{for } t = 10^4 + 1, \ldots, 2 \times 10^4, \end{cases} \tag{5.2}$$

where t is the index of the sequence $\underline{\mathbf{y}}$.

For the simulated data, we run a number of Gibbs samplers starting at different initial values, for M number of iterations where M is taken to be 5,000. Table 5.4 shows the simulation results for the mixture setting where number of components and number of states are 2 and 2, without segmentation. In Table 5.3 and Table 5.4, $\hat{\gamma}_{i,i,j}$ denotes the estimated transition probability from state i to itself in the j^{th} mixture component and $\hat{\xi}_{l,i,j}$ represents the estimated probability of emitting the symbol l in state i in the j^{th} mixture component for all i, j and l. In Table 5.3, $\hat{\boldsymbol{\lambda}}_j$ denotes the estimated vector of parameters required to specify the mixing weights associated with the j^{th} mixture component.

Table 5.3: Three runs of the Gibbs sampler for the simulated data modeled as a mixture of HMMs (without segmentation) along with the true parameters. Estimation is based on 5,000 iterations. The values in parenthesis in each column are the starting values corresponding to the estimates to their left.

$k = 2$, m=2, $r = 4$ and $L = 1$				
Runs	1	2	3	True Parameters
$\hat{\gamma}_{1,1,1}$	0.79 (0.37)	0.81 (0.23)	0.80 (0.90)	0.85
$\hat{\gamma}_{2,2,1}$	0.80 (0.19)	0.87 (0.57)	0.85 (0.37)	0.99
$\hat{\xi}_{1,1,1}$	0.24(0.41)	0.33 (0.31)	0.22 (0.28)	0.33
$\hat{\xi}_{2,1,1}$	0.19(0.09)	0.26 (0.03)	0.18 (0.07)	0.16
$\hat{\xi}_{3,1,1}$	0.21(0.10)	0.13 (0.24)	0.30(0.44)	0.16
$\hat{\xi}_{1,2,1}$	0.24(0.27)	0.27 (0.37)	0.28 (0.25)	0.27
$\hat{\xi}_{2,2,1}$	0.31(0.38)	0.26 (0.21)	0.28 (0.01)	0.23
$\hat{\xi}_{3,2,1}$	0.10(.26)	0.09 (0.14)	0.07 (0.27)	0.1
$\hat{\gamma}_{1,1,2}$	0.53(0.91)	0.35 (0.76)	0.67 (0.15)	0.6
$\hat{\gamma}_{2,2,2}$	0.51(0.47)	0.71 (0.47)	0.35 (0.27)	0.5
$\hat{\xi}_{1,1,2}$	0.16(.31)	0.19 (0.27)	0.10 (0.37)	0.2
$\hat{\xi}_{2,1,2}$	0.29(.20)	0.39 (0.24)	0.42 (0.14)	0.4
$\hat{\xi}_{3,1,2}$	0.19(.19)	0.19 (0.16)	0.22 (0.29)	0.2
$\hat{\xi}_{1,2,2}$	0.22(0.36)	0.21 (0.33)	0.30 (0.32)	0.15
$\hat{\xi}_{2,2,2}$	0.25(0.22)	0.21 (0.28)	0.11 (0.26)	0.25
$\hat{\xi}_{3,2,2}$	0.23(0.08)	0.17 (0.11)	0.24 (0.18)	0.2
$\hat{\boldsymbol{\lambda}}_1$	-0.43	0.03	0.72	
	0.22	-0.71	-0.79	

The Metropolis acceptance rate for $\hat{\boldsymbol{\lambda}}_1$ in the three runs are 0.01, 0.02 and 0.04.

5.0.3 The effect of Segmentation

Segmentation improves the parameter estimates which in turn results in better forecasts. Different segment lengths are considered and the simulation results when $L = 10$ and $L = 20$

are obtained. Simulation results corresponding to $L = 1$ and $L = 10$ are shown in Table 5.3 and Table 5.4 respectively. We use the same starting values and number of iterations for both $L = 1$ and $L = 10$. The mixing weights against (rescaled) sequence-index (or segments) for two runs of the Gibbs sampler without(left) and with segmentation (right) are displayed in Figure 5.3, when $L = 10$. The figure indicates that segmentation leads to better mixing weights in the two runs.

In Table 5.4, $\hat{\boldsymbol{\omega}}_j$ denotes the estimated vector of parameters required to specify the mixing

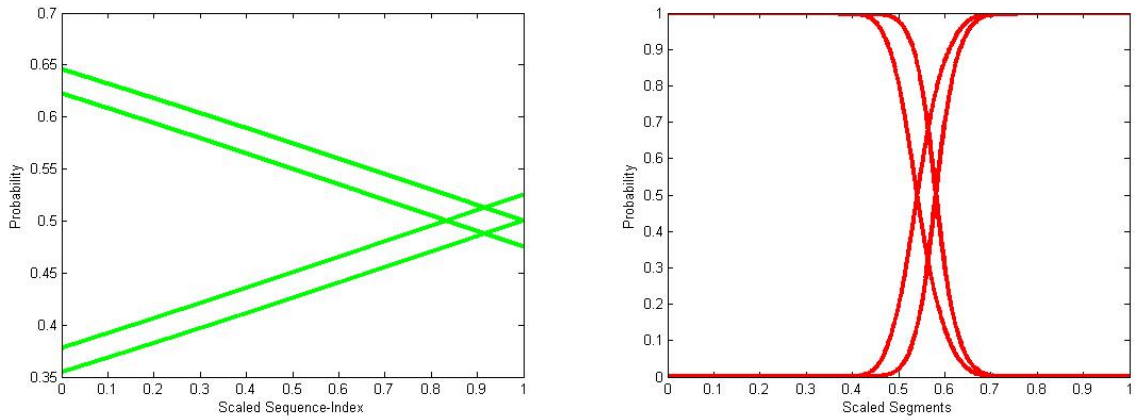

Figure 5.3: The mixing weights for the simulated data without segmentation (left) and with segmentation (right), when $L = 10$.

weights associated with the j^{th} mixture component.

Table 5.4: Three runs of the Gibbs sampler for the simulated data modeled as a mixture of HMMs (with segmentation) along with the true parameters. Estimation is based on 5,000 iterations. The values in parenthesis in each column are the starting values corresponding to the estimates to their left.

$k = 2$, m=2, $r = 4$ and $L = 10$				
Runs	1	2	3	True Parameters
$\hat{\gamma}_{1,1,1}$	0.79 (0.37)	0.85 (0.23)	0.87 (0.90)	0.85
$\hat{\gamma}_{2,2,1}$	0.95 (0.19)	0.93 (0.57)	0.92 (0.37)	0.99
$\hat{\xi}_{1,1,1}$	0.29(0.41)	0.29 (0.31)	0.27 (0.28)	0.33
$\hat{\xi}_{2,1,1}$	0.17(0.09)	0.17 (0.03)	0.27 (0.07)	0.16
$\hat{\xi}_{3,1,1}$	0.18(0.10)	0.19 (0.24)	0.14(0.44)	0.16
$\hat{\xi}_{1,2,1}$	0.26(0.27)	0.24 (0.37)	0.24 (0.25)	0.27
$\hat{\xi}_{2,2,1}$	0.23(0.38)	0.26 (0.21)	0.27 (0.01)	0.23
$\hat{\xi}_{3,2,1}$	0.11(.26)	0.12 (0.14)	0.14 (0.27)	0.1
$\hat{\gamma}_{1,1,2}$	0.53(0.91)	0.53 (0.76)	0.8 (0.15)	0.6
$\hat{\gamma}_{2,2,2}$	0.54(0.47)	0.51 (0.47)	0.75 (0.27)	0.5
$\hat{\xi}_{1,1,2}$	0.19(.31)	0.3 (0.27)	0.19 (0.37)	0.2
$\hat{\xi}_{2,1,2}$	0.38(.20)	0.32 (0.24)	0.34 (0.14)	0.4
$\hat{\xi}_{3,1,2}$	0.21(.19)	0.27 (0.16)	0.16 (0.29)	0.2
$\hat{\xi}_{1,2,2}$	0.21(0.36)	0.10 (0.33)	0.22 (0.32)	0.15
$\hat{\xi}_{2,2,2}$	0.33(0.22)	0.31 (0.28)	0.25 (0.26)	0.25
$\hat{\xi}_{3,2,2}$	0.15(0.08)	0.16 (0.11)	0.25 (0.18)	0.2
$\hat{\boldsymbol{\omega}}_1$	35.97	-39.32	37.97	
	-66.49	50.62	-65.37	

The Metropolis acceptance rate for $\hat{\boldsymbol{\omega}}_1$ in the three runs are 0.89, 0.87 and 0.87.

For each segment length, we compute the h-step forecasting distributions, where $h = 1, 2, 3$. For each value of h, we compare the forecast distribution when $L = 1$ with the ones corrresponding to $L = 10$ and $L = 20$. Note that $L = 1$ means that there is no

segmentation. For a given value of h, we compute the mean squared error (MSE) of the forecast distributions corresponding to $L = 1$, $L = 10$ and $L = 20$. The MSEs corresponding to $L = 10$ and $L = 20$ are compared to that corresponding to $L = 1$. For the given values of h, Table 5.5 shows a significant reduction in MSE of forecast distributions as a result of segmentation.

Table 5.5: The mean squared error (MSE) of forecast distributions corresponding to $L = 1$, $L = 10$ and $L = 20$

MSE			
h	L=1	L=10	L=20
1	1.33e-05	3.82e-06	3.83e-06
2	1.17e-05	5.82e-06	4.03e-06
3	1.14e-05	5.63e-06	3.88e-06

For a given value h, we compute the MSE for the forecast distributions corresponding to a single HMM and that corresponding to the mixtures of HMMs, when $L = 10$. The MSE for the mixtures is found to be smaller.

Note that there is no space between the pink and black bars, that is, they both represent a

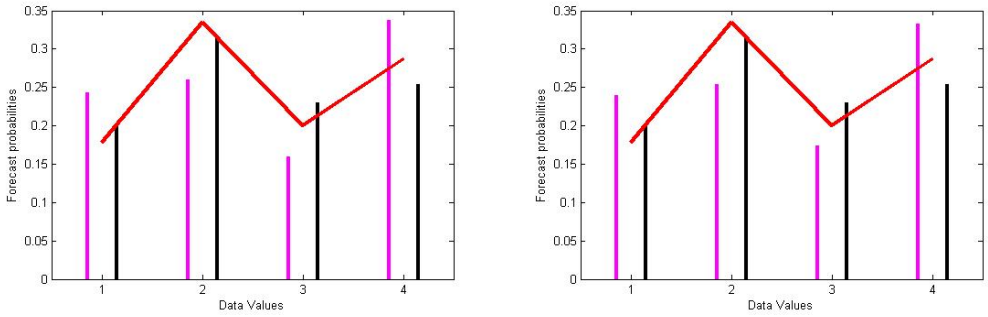

Figure 5.4: Plots of forecast distributions for $h = 1$ (left) and h=4 (right). The pink bars are those corresponding to $L = 1$ and the black bars are those corresponding to $L = 10$. The continuous (red) line shows the true forecast distributions.

particular data value. As you see in Figure 5.4 the black bars are closer to the true forecasts than the pink ones, implying that segmentation improves forecast distributions.

Références

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities* **3,** 1–8.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics* **41,** 164–171.

Boys, R. J., Henderson, D., and Wilkinson, D. J. (2000). Detecting homogeneous segments in DNA sequences by using hidden Markov models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **49,** 269–285.

Boys, R. J. and Henderson, D. A. (2004). A Bayesian approach to DNA sequence segmentation. *Biometrics* **60,** 573581.

Brooks, S. P. (1998). Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society. Series D (The Statistician)* **47,** 69–100.

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference In Hidden Markov Models.* Springer Series in Statistics. Springer Science + Business Media, Inc.

Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics* **75,** 79–97.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journalof the American Statistical Association* **91,** 883–904.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39,** 1–38.

Ephraim, Y. and Merhav, N. (2002). Hidden Markov processes. *IEEE Trans. Inform. Theory* **48,** 1518–1569.

Felsenstein, J. and Churchill, G. (1996). A hidden Markov model approach to variation among sites in rate of evolution. *Mol. Biol. Evol* **13(1),** 93–104.

Geman, D. and Geman, S. (1983). Parameter estimation for some Markov random fields. Technical report, Reports on Pattern Analysis No. 11, Division of Applied Mathematics, Brown University.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE-PAMI* **6,** 721–741.

He, Y. and Kundu, A. (1991). 2-D shape classification using hidden Markov model. *IEEE Transactions on Pattern Analysis Machine Intelligence* **13,** 1172–1184.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local- experts. *Neural Computation* **3,** 79–87.

Juang, B. H. and Rabiner, L. R. (1991). Hidden Markov models for speech recognition. *Technometrics* **33,** 251–272.

Leroux, B. and Puterman, M. (1992). Maximum penalized likelihood estimation for independent and Markov-Dependentpoisson mixtures. *Biometrics* **48,** 545 – 558.

Marin, J. and Robert, C. (2007). *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer, New York.

Mcgrory, C. A. and Titterington, D. M. (2009). Variational Bayesian analysis for hidden Markov models. *Australian & New Zealand Journal of Statistics* **51(2),** 227–244.

Peng, F., Jacobs, R. A., and Tanner, M. A. (1996). Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journalof the American Statistical Association* **91,** 953–960.

Qian, W. and Titterington, D. M. (1991). Estimation of parameters in hidden Markov models. *Philosophical Transactions: Physical Sciences and Engineering* **337,** 407–428.

Rabiner, R. L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE* **77,** 257–286.

Robert, C., Celeux, G., and Deibolt, J. (1993). Bayesian estimation of hidden Markov chains: A stochastic implementation. *Statistics & Probability Letters* **16,** 77–83.

Ross, S. (1996). *Stochastic Processes*. John Wiley & Sons, Inc.

Ryden, T. and Titterington, D. M. (1998). Computational Bayesian analysis of hidden

Markov models. *Journal of Computational and Graphical Statistics* **7,** 194–211.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association* **00,**.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* **82,** 528550.

Welch, L. R. (2003). Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter* **53,** 1–13.

Wood, S., Rosen, O., and Kohn, R. (2011). Bayesian mixtures of autoregressive models. *J. of Computational and Graphical Statistics* **20,** 174–195.

Zucchini, W. and MacDonald, I. (2009). *Hidden Markov Models for Time Series*. Chapman & Hall, CRC Press.

Appendix A

Matlab Code for Estimating the Parameters of the j^{th} Mixture-Component

```
% Given the transition and emission parameters at the previous iteration,
% this function computes and returns those at the current iteration.

function [tr,em]  = gibbsj(seq,trans,emiss)
n=length(seq);
data=zeros(4,n);
for i=1:n
if  (seq(i)==4)
    data(:,i)=[0,0,0,1];
elseif  (seq(i)==3)
    data(:,i)=[0,0,1,0];
elseif (seq(i)==2)
    data(:,i)=[0,1,0,0];
else
    data(:,i)=[1,0,0,0];
end
end
```

```
%Initialize parameter values

z=zeros(1,n);
pz=zeros(2,n);
for t=1:n
     u=unifrnd(0,1);
    if(t==1)
         pz(1,t)=trans(1,1)* emiss(1,seq(t));
         pz(2,t)=trans(2,2)* emiss(2,seq(t));
         if(u<(pz(1,t)/sum(pz(:,t))))
             z(t)=0;
         else
             z(t)=1;
         end
    else
           pz(1,t)= trans(z(t-1)+1,1)*emiss(1,seq(t));
           pz(2,t)= trans(z(t-1)+1,2)*emiss(2,seq(t));
            if(u<(pz(1,t)/sum(pz(:,t))))
                 z(t)=0;
            else
                z(t)=1;
            end
    end
end
         % Calculate the necessary sufficient statistics(mij's)

         count11=0; count12=0; count21=0; count22=0;
         for i=1:n-1
```

```
        if(z(i)==0&& z(i+1)==0)
            count11=count11+1;
        elseif(z(i)==0&& z(i+1)==1)
            count12=count12+1;
        elseif(z(i)==1&& z(i+1)==0)
            count21=count21+1;
        else count22= count22+1;
        end
    end
    % Calculate the necessary sufficient statistics(nij's)

  sum1=[0,0,0,0]; sum2=[0,0,0,0];
    for i=1:n
        if(z(i)==0)
            sum1=sum1+data(:,i)';
        else
            sum2=sum2+data(:,i)';
        end
    end
a=[1,1;1,1];b=[1,1,1,1;1,1,1,1];

    %Generate transitions from drichlet posterior

    d1=gamrnd([a(1,1)+ count11,a(1,2)+ count12],1);
    d2=gamrnd([a(2,1)+ count21,a(2,2)+ count22],1);
    trans=[d1./sum(d1);d2./sum(d2)];

    %Generate emissions from drichlet posterior
```

```
        d1=gamrnd([b(1,1)+ sum1(1),b(1,2)+  sum1(2),b(1,3)+sum1(3),b(1,4)+sum1(4)],1);
        d2=gamrnd([b(2,1)+ sum2(1),b(2,2)+ sum2(2),b(2,3)+sum2(3),b(2,4)+sum2(4)],1);
        emiss=[d1./sum(d1);d2./sum(d2)];

tr=trans;
em=emiss;
```

Appendix B

Matlab Code for Estimating Parameters and Forecasting Distributions of a Single HMM

```
function [avgtr,avgem, averageforecast]  = gibbssampler(seq)

% This function returns estimates of parameters and forecast distributions
% of a single HMM for a given sequence.

n=length(seq);
data=zeros(4,n);
for i=1:n
if  (seq(i)==4)
    data(:,i)=[0,0,0,1];
elseif  (seq(i)==3)
    data(:,i)=[0,0,1,0];
elseif (seq(i)==2)
    data(:,i)=[0,1,0,0];
else
    data(:,i)=[1,0,0,0];
end
end
```

```
%Initialize parameter values

iterations=10000;
warmup=2001;
z=zeros(1,n);
trans=zeros(2,2,iterations);
emiss=zeros(2,4,iterations);
d1=gamrnd([1,1],1);
d2=gamrnd([1,1],1);
trans(:,:,1)=[d1./sum(d1);d2./sum(d2)];
d1=gamrnd([1,1,1,1],1);
d2=gamrnd([1,1,1,1],1);
emiss(:,:,1)=[d1./sum(d1);d2./sum(d2)];
pz=zeros(2,n);
for t=1:n
     u=unifrnd(0,1);
    if(t==1)
         pz(1,t)=trans(1,1,1)* emiss(1,seq(t),t);
         pz(2,t)=trans(2,2,1)* emiss(2,seq(t),t);
         if(u<(pz(1,t)/sum(pz(:,t))))
             z(t)=0;
         else
             z(t)=1;
         end
    else
           pz(1,t)= trans(z(t-1)+1,1,1)*emiss(1,seq(t),1);
           pz(2,t)= trans(z(t-1)+1,2,1)*emiss(2,seq(t),1);
```

```
            if(u<(pz(1,t)/sum(pz(:,t))))
                 z(t)=0;
            else
                z(t)=1;
            end
    end
end

          % Calculate the necessary sufficient statistics(mij's)

          count11=0; count12=0; count21=0; count22=0;
          for i=1:n-1
              if(z(i)==0&& z(i+1)==0)
                  count11=count11+1;
              elseif(z(i)==0&& z(i+1)==1)
                  count12=count12+1;
              elseif(z(i)==1&& z(i+1)==0)
                  count21=count21+1;
              else count22= count22+1;
              end
          end

          % Calculate the necessary sufficient statistics(nij's)

        sum1=[0,0,0,0]; sum2=[0,0,0,0];
          for i=1:n
              if(z(i)==0)
                  sum1=sum1+data(:,i)';
```

```
          else
              sum2=sum2+data(:,i)';
          end
      end
 a=[1,1;1,1];b=[1,1,1,1;1,1,1,1];

 % Iterations

for m=1:iterations

      %Generate transitions from drichlet posterior.

      d1=gamrnd([a(1,1)+ count11,a(1,2)+ count12],1);
      d2=gamrnd([a(2,1)+ count21,a(2,2)+ count22],1);
      trans(:,:,m)=[d1./sum(d1);d2./sum(d2)];

      %Generate emissions from drichlet posterior.

      d1=gamrnd([b(1,1)+ sum1(1),b(1,2)+ sum1(2),b(1,3)+sum1(3),b(1,4)+sum1(4)],1);
      d2=gamrnd([b(2,1)+ sum2(1),b(2,2)+ sum2(2),b(2,3)+sum2(3),b(2,4)+sum2(4)],1);
      emiss(:,:,m)=[d1./sum(d1);d2./sum(d2)];
      u1=unifrnd(0,1);
      u2=unifrnd(0,1);
     met_rat=  u1/ pz(mod(z(1)+1,3),1);
     acc=min(1,met_rat);
      if (u2 < acc)
          z(1)=mod(z(1)+1,2);
      end
```

```
 for t=1:n-1
     u=unifrnd(0,1);
    if(t==1)
         pz(1,t)=trans(1,1,m)* emiss(1,seq(t+1),m)*trans(1,z(2)+1,m);
         pz(2,t)=trans(2,2,m)* emiss(2,seq(t+1),m)*trans(2,z(2)+1,m);
         if(u<(pz(1,t)/sum(pz(:,t))))
             z(t)=0;
         else
             z(t)=1;
         end
    else
           pz(1,t)= trans(z(t-1)+1,1,m)*emiss(1,seq(t),m)*trans(1,z(t+1)+1,m);
           pz(2,t)= trans(z(t-1)+1,2,m)*emiss(2,seq(t),m)*trans(2,z(t+1)+1,m);
            if(u<(pz(1,t)/sum(pz(:,t))))
                 z(t)=0;
            else
                z(t)=1;
            end
    end
end
 % Calculate the necessary sufficient statistics(mij's)

         count11=0; count12=0; count21=0; count22=0;
         for i=1:n-1
             if(z(i)==0 && z(i+1)==0)
                 count11=count11+1;
             elseif(z(i)==0 && z(i+1)==1)
                 count12=count12+1;
```

```
            elseif(z(i)==1&& z(i+1)==0)
                count21=count21+1;
            else count22= count22+1;
            end
        end

        % Calculate the necessary sufficient statistics(nij's)

       sum1=[0,0,0,0]; sum2=[0,0,0,0];
        for i=1:n
            if(z(i)==0)
                sum1=sum1+data(:,i)';
            else
                sum2=sum2+data(:,i)';
            end
        end
             [~,~,ff1,~,~]=hmmdecode(seq,trans(:,:,m),emiss(:,:,m));
     tr1=trans(:,:,m)^h;
     fore1=zeros(1,4);
    for j=1:4
        fore1(j)=ff1(:,end)'*tr1*diag(emiss(:,j,m))*one';
    end
     forecast(m,:)=fore1;
  end
avgtr=mean(trans(:,:,warmup:end),3);
avgem=mean(emiss(:,:,warmup:end),3);
averageforecast=mean(forecast(warmup:end,:),1);
```

Appendix C

Matlab Code for Estimating Parameters of Mixture of HMMs

```
function [weight_fns,acceptance, delta1, tr1, em1, tr2, em2,averageforecast]
 = segmentation(seq,segment_len,  jj)

% Given a sequence, segment length and the number of mixture components,
% this function returns estimates of  the weights functions, covariates of
%the mixing weights, transition probability matrix and emission parameters
%of each of the mixture components and h-step forecasting distributions for
% h=1, 2, ....The variables are self-explanatory. Use segment length=1, if
% you want to run the code without segmentation.



ahead=5;  %forecast-horizon
one=[1 1];

% Vague hyperparameter for the covariates of mixing weghts.

var_prior_delta=1000;
counter=0;
n=length(seq);
data=zeros(4,n);
```

```
% Multinomial representation of the data.

for i=1:n
if  (seq(i)==4)
    data(:,i)=[0,0,0,1];
elseif  (seq(i)==3)
    data(:,i)=[0,0,1,0];
elseif (seq(i)==2)
    data(:,i)=[0,1,0,0];
else
    data(:,i)=[1,0,0,0];
end
end

%Initialization step

iterations=5000;
%Burn in period

warmup=1001;
num_segments=n/segment_len;
forecast=zeros(fn,4,iterations);
trans1=zeros(2,2,iterations);
trans2=zeros(2,2,iterations);
emiss1=zeros(2,4,iterations);
emiss2=zeros(2,4,iterations);
delta=zeros(2,jj,iterations);
```

```
d1=gamrnd([10,1],1);
d2=gamrnd([1,10],1);
trans1(:,:,1)=[d1./sum(d1);d2./sum(d2)];
d1=gamrnd([10,1],1);
d2=gamrnd([1,10],1);
trans2(:,:,1)=[d1./sum(d1);d2./sum(d2)];
d1=gamrnd([1,1,1,1],1);
d2=gamrnd([1,1,1,1],1);
emiss1(:,:,1)=[d1./sum(d1);d2./sum(d2)];
d1=gamrnd([1,1,1,1],1);
d2=gamrnd([1,1,1,1],1);
emiss2(:,:,1)=[d1./sum(d1);d2./sum(d2)];
p=zeros(jj,num_segments,iterations);
pz=zeros(jj,num_segments);
wt1=zeros(1,n);
wt2=zeros(1,n);
wt11=zeros(1,num_segments);
wt22=zeros(1,num_segments);
u1=rand(1,num_segments);
u2=rand(1,num_segments);
z=u1<u2;
indd=zeros(1,num_segments);
indd(z)=1;
indd(indd==0)=2;
z_1=ones(2,num_segments);
z_1(2,:)= (1:num_segments)/num_segments;
[delta_mean , ~, stats]= mnrfit(z_1(2,:),indd);
delta(:,1:jj-1,1)=delta_mean;
```

```
%Iteration step

for m =2:iterations
    for j=1:jj
        p(j,:,m-1)=exp(delta(:,j,m-1)'*z_1)./sum(exp(delta(:,:,m-1)'*z_1),1);
    end

% compute the prob. of zij's

    u=rand(1,num_segments);
    [~,~,Forward1,~,S1] = hmmdecode(seq,trans1(:,:,m-1),emiss1(:,:,m-1));
    Forward1=Forward1.*[S1;S1];
    f1=sum(Forward1(:,2:end));
    wt1(1)=sum(Forward1(:,1));
    wt1(2:end)=f1(2:end)./f1(1:end -1);
    [~,~,Forward2,~,S2] = hmmdecode(seq,trans2(:,:,m-1),emiss2(:,:,m-1));
    Forward2=Forward2.*[S2;S2];
    f2=sum(Forward2(:,2:end));
    wt2(1)=sum(Forward2(:,1));
    wt2(2:end)=f2(2:end)./f2(1:end -1);
    i=1;j=1;
    while(i<=n && j<=num_segments)
        wt11(j)= prod(wt1(i:i+segment_len-1));
        wt22(j)= prod(wt2(i:i+segment_len-1));
        i=i+segment_len;
        j=j+1;
    end
```

```
pz=p(:,:,m-1).*[wt11;wt22];
pz=pz./[sum(pz);sum(pz)];
z=u<pz(1,:);
indices1=find(z);
indices2=find(~z);
n1=size(indices1);
n2=size(indices2);
n1=n1(2);
n2=n2(2);
starting_ind1=1+(indices1-1)*segment_len;
starting_ind2=1+(indices2-1)*segment_len;
seq11=zeros(1,n1*segment_len);
seq22=zeros(1,n2*segment_len);
if(~isempty(indices1))
    i=1;j=1;
    while(i<=n && j<=n1)
        seq11(i:i+segment_len-1)=seq(starting_ind1(j):segment_len*indices1(j));
        j=j+1;
        i=i+segment_len;
    end
end
if(~isempty(indices2))
    i=1;j=1;
    while(i<=n && j<=n2)
        seq22(i:i+segment_len-1)=seq(starting_ind2(j):segment_len*indices2(j));
        j=j+1;
        i=i+segment_len;
    end
```

```
 end
 indd=zeros(1,num_segments);
 indd(z)=1;
 indd(indd==0)=2;
 %compute sufficient statistics
 ind=[z;~z];
 [trans1(:,:,m),emiss1(:,:,m)]  = gibbsj(seq11,trans1(:,:,m-1),emiss1(:,:,m-1));
 [trans2(:,:,m),emiss2(:,:,m)]  = gibbsj(seq22,trans2(:,:,m-1),emiss2(:,:,m-1));
 [delta_mean , ~, stats]= mnrfit(z_1(2,:),indd);
 delta_var=stats.covb;

% Metropolis Hastings for delta

 delta_prior_var=var_prior_delta*eye(2);
 delta_prop=mvnrnd(delta_mean,delta_var)';
 delta(:,1:jj-1,m)=1.2*delta_prop;
 delta_curr=delta(:,1:jj-1,m-1);

 %Evaluating the proposal density at the current and proposed values

  log_prop_delta_prop=-0.5*(delta_prop-delta_mean)'
   *(delta_var\(delta_prop-delta_mean));
  log_prop_delta_curr=-0.5*(delta_curr-delta_mean)'
   *(delta_var\(delta_curr-delta_mean));

 %Evaluting the likelihood at the current and proposed values

 loglike_curr=sum(sum(ind.*(delta(:,:,m-1)'*z_1),1)
```

```
 -log(sum(exp(delta(:,:,m-1)'*z_1),1)));
loglike_prop=sum(sum(ind.*(delta(:,:,m)'*z_1),1)
 -log(sum(exp(delta(:,:,m)'*z_1),1)));

%Evaluting the prior density at  at the current and proposed values

 log_prior_prop=-0.5*delta_prop'*(delta_prior_var\delta_prop);
 log_prior_curr=-0.5*delta_curr'*(delta_prior_var\delta_curr);
 % The posterior as the sum of the likelihood and prior
  log_target_prop=loglike_prop+log_prior_prop;
  log_target_curr=loglike_curr+log_prior_curr;

 %Determining acceptance propbability

  u=rand;

acc(m-1)=min(1,exp(log_target_prop-log_target_curr
 +log_prop_delta_curr-log_prop_delta_prop));
      if(u>acc(m-1))
           delta(:,1:jj-1,m)=delta(:,1:jj-1,m-1);
           counter=counter+1;
      end
[~,~,ff1,~,~]=hmmdecode(seq11,trans1(:,:,m-1),emiss1(:,:,m-1));
[~,~,ff2,~,~]=hmmdecode(seq22,trans2(:,:,m-1),emiss2(:,:,m-1));
 for h=1:ahead
tr1=trans1(:,:,m-1)^h;
tr2=trans2(:,:,m-1)^h;
forecast1=zeros(1,4);
```

```
    forecast2=zeros(1,4);
    for j=1:4
        forecast1(j)=ff1(:,end)'*tr1*diag(emiss1(:,j,m-1))*one';
        forecast2(j)=ff2(:,end)'*tr2*diag(emiss2(:,j,m-1))*one';
    end
    pt=exp(delta(:,:,m-1)'*[1;(n+1)/n])./sum(exp(delta(:,:,m-1)'*[1;(n+1)/n]),1);
    forecast(h,:,m-1)= pt(1)*forecast1 + pt(2)*forecast2;
     end
end

  weight_fns=mean(p(:,:,warmup:end),3);
  acceptance=mean(acc(warmup:end));
  delta1= mean(delta(:,:,warmup:end),3);
  tr1=mean(trans1(:,:,warmup:end),3);
  tr2=mean(trans2(:,:,warmup:end),3);
  em1=mean(emiss1(:,:,warmup:end),3);
  em2=mean(emiss2(:,:,warmup:end),3);
  averageforecast=mean(forecast(:,:,warmup:end),3);
```

Appendix D

Matlab Code for plotting Mixing weights

```
[]=function plotting(weight_fns1, weight_fns2, segment_len)
% Code for plotting Mixing weights

n=length(weight_fns1);
num_segments=n/segment_len;
z_1=ones(2,num_segments);
z_1(2,:)= (1:num_segments)/num_segments;
p=plot(z_1(2,:),weight_fns1(2,:),z_1(2,:),weight_fns1(1,:))
xlabel('Scaled Segments')
ylabel('Probability')
% Change the line color to red and
% set the line width to 2 points
set(p,'Color','red','LineWidth',3)
hold on;
p=plot(z_1(2,:),weight_fns2(2,:),z_1(2,:),weight_fns2(1,:))
xlabel('Scaled Segments')
ylabel('Probability')
% Change the line color to red and
% set the line width to 3 points
```

```
set(p,'Color','red','LineWidth',3)
```

Curriculum Vitae

Samson Ghebremariam was born on January 1, 1983. The first son of Laine Ghebremariam and Mebrat Mesfin, he graduated from Saint George High School, Mendefera, Eritrea, in the spring of 2001. He entered Asmara University in the fall of 2001, graduated with honors and recived his bachelor's degree in Statistics and Demography minoring in Computer Science in the spring of 2005. While pursuing his bachelor's degree in Statistics he worked as a statistician, and as a programmer at the Testing Center and Registrar office, Asmara University. In the fall of 2005 he was assigned to the Eritrea Institute of Technology where he taught statistics and computer programming courses for about four years.

In the fall of 2009, he entered the Graduate School of The University of Texas at El Paso. While pursuing a master's degree in Statistics he worked as a Teaching Assistant in the department of Mathematical Sciences at the University of Texas at El Paso, and as a part-time instructor at Western Technical College, El Paso. He recieved his master's degree in Statistics in the summer of 2011. In the fall of 2011 he will join the University of Florida to pursue his PhD in Biostatistics.

Current address: 806 W. Yandell Dr.

El Paso, Texas 79902