Alignment of Time Course Microarray Data with Hidden Markov Models

Sean Robinson

Primary Supervisor: Associate Professor Gary Glonek
Secondary Supervisor: Associate Professor Inge Koch

Declaration

I, Sean Robinson certify that this work contains no material that has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University’s digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed by student:

Date:

Signed by supervisor:

Date:
Acknowledgements

I would like to extend a huge thank you to Gary and Inge for their supervision during this project. Particular thanks are also due to Brian Danilko for help with software, Chris Davies for help with the data, Jess Kasza for help with math, and Trent Mattner and Matt Roughan for their help as postgraduate coordinators.
Abstract

Time course microarray experiments allow for insight into biological processes by quantifying changes in gene expression over a time period of interest. This project is motivated by time course microarray data from an experiment conducted on grapevines over the development cycle of the grape berries at a number of different vineyards in South Australia. Although the underlying biological process is the same at each vineyard, there are differences in the timing of the development cycle at different vineyards due to local conditions.

The aim of this project is to construct a methodology to align the data from different vineyards in order to obtain a common representation of the gene expression over the development cycle of the grape berries for each gene. Hidden Markov models (HMMs) have been used to model time series data in a number of domains and have also been used to model time course microarray data. We review these applications in addition to the use of HMMs for particular alignment problems in genomic sequence data. We present an extension of HMMs and propose a novel alignment methodology based on this extension. We evaluate the proposed alignment methodology by applying it to simulated data prior to using it to align the grapevine data.
Contents

1 Introduction 1
   1.1 Motivation 3
   1.2 Aims 6
   1.3 Overview of Thesis 6

Part I

2 Notation, Preliminary Definitions and Results 9
   2.1 Random Variables and Independence 9
   2.2 Sequences of Random Variables 12
   2.3 Graphical Models 13

3 Hidden Markov Models 16
   3.1 Definition 16
   3.2 Parameterisation 18

4 Associated Problems for HMMs 20
   4.1 Evaluation of the Marginal Emission Density 21
   4.2 Most Likely Corresponding Realised State Sequence 26
   4.3 Maximum Likelihood Estimator of the Parameters 29

5 Alignment with HMMs 34
   5.1 Pair HMMs for Alignment of Genomic Sequences 34
   5.2 Extensions of Pair HMMs 37
      5.2.1 Binary Markov Dynamics with Continuous Emissions 37
      5.2.2 Additional Information Incorporated into Model 39
5.3 Simple Alignment with an HMM

Part II

6 L(t)-fold HMMs

6.1 Definition

6.2 Parameterisation

7 Associated Problems for L(t)-fold HMMs

7.1 Evaluation of the Marginal Emission Density

7.2 Most Likely Corresponding Realised State Sequence

7.3 Maximum Likelihood Estimator of the Parameters

8 Alignment with L(t)-fold HMMs

8.1 An Example Model

8.2 Model for the Grapevine Data

9 Model Fitting Methodology

9.1 Model Fitting Algorithm

9.2 Simulation Example

10 Alignment of the Grapevine Data

10.1 Fitting the Model

10.2 Common Representations

11 Alignment Diagnostics

11.1 Ordering Based on Log-likelihood

11.2 Fitting an HMM and the Estimated HMM Parameters

11.2.1 Comparison of Estimated Parameters

11.2.2 Difference in Log-likelihood

12 Conclusion

12.1 Summary of Thesis

12.2 Discussion
Appendices

A  The Grapevine Data  

B  The EM Algorithm  

C  Proof of Theorem 4.6  

Bibliography
Chapter 1

Introduction

Genes can be considered as sections of the genome that convey genetic information (Watson, Baker, Bell, Gann, Levine & Losick 2008). The genome of an organism is encoded in the form of deoxyribonucleic acid (DNA) and within living cells, DNA is transcribed into messenger ribonucleic acid (mRNA), which is then translated into a gene product, typically protein. This transcription and translation process is known as gene expression and the gene expression level of a gene can be measured by the abundance of corresponding mRNA transcripts.

Microarray experiments are a well established method for simultaneously measuring multiple gene expression levels across a genome and data are collected for thousands of known genes in a single experiment. When gene expression levels are measured on the same material at subsequent time points, the experiment is known as a time course and the resulting data set is known as time course microarray data (Speed 2003). For an individual gene, the expression level as measured over time is known as an expression profile or just a profile. Time course microarray experiments enable the changes in gene expression to be quantified over a time period of interest and the analysis of time course microarray data allows for insight into biological processes.

The use of statistical analysis is necessary for many problems in time course microarray experiments (Speed 2003). Problems preceding modelling the data include the design of experiments and pre-processing the data, which require statistical techniques such as image analysis, outlier detection and noise correction. Once time course microarray data are available for analysis, current areas of investigation include clustering, classification and modelling gene regulatory networks (Solomon 2009).

We consider a time course microarray experiment conducted by Dr Chris Davies, Senior Re-
Figure 1.1: Expression profiles from the Willunga vineyard. Left to right, top to bottom, the profiles correspond to genes with probe identification 1621649_at, 1610245_at, 1616418_at and 1609985_at.
search Scientist, CSIRO Plant Industry Research Division. We only provide the necessary detail to motivate our problem here, further details of the experiment are given in Appendix A. The experiment was conducted on grapevines at a number of different vineyards in South Australia. The species and variety of the grapevines were the same at all vineyards and gene expression levels were measured weekly at each vineyard. The experiment was run over the duration of the development cycle of the grape berries, roughly an 18 week process from the closed-flower to ripe-red stage of the berries, which varied between vineyards. In this project we are not interested in amplitude of expression level but rather in the changes in gene expression over time and hence the shape of the expression profiles. For this reason, we scale the data so that all observed expression levels lie in the interval \([0,1]\) (Appendix A). Observed expression profiles from the Willunga vineyard can be seen in Figure 1.1.

We aim to combine the multiple expression profiles from different vineyards in order to obtain a single profile that represents the relevant gene expression information over the development of the grape berries for each gene. By combining the data from multiple vineyards we obtain more precise estimates of the gene expression over the development of the grape berries, rather than just using the data from a single vineyard. The common representations could then be used for cluster analysis for example, or for other analysis of time course microarray data as discussed by Solomon (2009).

In order to combine the data from multiple vineyards requires an alignment methodology, explained below. In general, the alignment of time course microarray data is an important problem since, as discussed by Aach & Church (2001):

> biological processes have the property that multiple instances of a single process may unfold at different and possibly non-uniform rates in different organisms, strains, individuals, or conditions.

Such different rates may affect the timing of gene expression, which will be manifest in the observed expression profiles. We consider the way in which this issue is present for us below.

### 1.1 Motivation

For this project we focus on two vineyards considered in the time course experiment that are situated in the Willunga and Clare grape growing regions of South Australia (Figure 1.2). Dr Davies selected 2062 genes for which we have a pair of expression profiles, one each from the
Willunga and Clare vineyards. The expression profiles corresponding to these 2062 genes form the data set we hereafter call the ‘grapevine data’ (Appendix A). For each pair of profiles in the grapevine data, we aim to obtain a single profile that captures the relevant gene expression information from both vineyards. Pairs of observed expression profiles from the Willunga and Clare vineyards can be seen in Figure 1.3.

We know that the rate of development of the grape berries was different at the Willunga and Clare vineyards. This is apparent in the different lengths of the development cycle of the grape berries, which was 19 weeks at the Willunga vineyard and 17 weeks at the Clare vineyard. As the experimental procedure called for weekly measurements to be conducted over the duration of the development cycle (Appendix A), the expression profiles from Willunga have length 19, while the expression profiles from Clare have length 17 (Figure 1.3). Hence we require an alignment between the pairs of different length profiles from the Willunga and Clare vineyards in order to obtain common representations.

The alignment methodology we develop will be based on an extension of a hidden Markov model. Hidden Markov models (HMMs) have been used for the analysis of time series data in a wide range of applications, including modelling time course microarray data (Bar-Joseph 2004). Additionally, HMM based methodologies have been extensively used for alignment problems in
Figure 1.3: Expression profiles from the Willunga (•) and Clare (⋆) vineyards. Left to right, top to bottom, the pairs of expression profiles correspond to genes with probe identification 1621649_at, 1610245_at, 1616418_at and 1609985_at.

The grapevine data set is made up of expression profiles corresponding to genes believed to exhibit similar gene expression at each vineyard (Appendix A) and hence each pair of observed expression profiles are expected to have the same basic shape. Cluster analysis had been carried out on the expression profiles from the Clare vineyard and the collection of genes selected for this project by Dr Davies correspond to 16 of the groups from that analysis. Plots of the grapevine data by group and vineyard are given in Appendix A where it can be seen that generally, the expression profiles appear to have the same basic shape at both vineyards in each group. This is clearly the case for the pairs of expression profiles in Figure 1.3, where each pair share the same basic shape, although not the fine details and where prominent features occur at different times. Hence these pairs of profiles are conducive to an alignment.

We know that there are genes that may not exhibit similar gene expression at the Willunga and Clare vineyards over the development of the grape berries. For instance, the expression of so called ‘heat shock’ genes is driven by factors that may have differed across vineyards in the time course experiment. For genes that do not exhibit similar gene expression, the corresponding expression profiles will not be conducive to an alignment and there are possibly such genes represented in the grapevine data. Hence we additionally aim to use the model to identify pairs of profiles for which the proposed alignment is inappropriate.

1.2 Aims

We aim to develop a methodology to:

1. Align the pairs of expression profiles from the Willunga and Clare vineyards;

2. Obtain a common representation of the pair of expression profiles for each gene; and

3. Identify pairs of profiles for which the proposed alignment is inappropriate.

1.3 Overview of Thesis

This thesis is divided into two parts. We first review the necessary background material and literature. Then we propose an extension to HMMs and present a novel alignment methodology based on the extension.
In Part I we present the notation, preliminary definitions and results needed for the rest of the thesis (Chapter 2). We define and describe HMMs (Chapter 3) and present the mathematical solutions to the three associated HMM problems so that the model can be used in practice (Chapter 4). We review the HMM based alignment methodologies for genomic sequences data and consider how such methods could be extended in order to achieve our particular aims (Chapter 5).

In Part II we propose an extension to HMMs (Chapter 6) and present mathematical solutions to the three analogous associated problems (Chapter 7). We consider the way in which an alignment of time course data can be achieved under the extended HMM and propose an alignment model for the grapevine data (Chapter 8). We present a model fitting procedure which we illustrate on simulated data (Chapter 9). We implement and evaluate our proposed alignment methodology on the grapevine data (Chapter 10) and consider alignment diagnostics under the model (Chapter 11). We conclude by giving a summary of how we achieved our aims and a final discussion of our methodology and key future work (Chapter 12).
Part I
Chapter 2

Notation, Preliminary Definitions and Results

We present the necessary notation, preliminary definitions and results that will enable us to define and describe hidden Markov models in addition to the extension. The entire content of this chapter is a review of the relevant material. We first consider random variables and, marginal and conditional independence (Section 2.1). We then define discrete time stochastic processes and Markov chains (Section 2.2). Finally we define graphical models and conditional independence graphs (Section 2.3).

2.1 Random Variables and Independence

We use capital letters to denote random variables and corresponding lowercase letters to denote realisations of random variables. A random variable $Z$ takes values in the space $\Omega_Z$ and we write the event ‘$Z$ takes the value $z$’ as $Z = z$.

We use $p$ to denote the probability density function of a continuous random variable or the probability mass function of a discrete random variable. We will use the argument of the density or mass function to identify the random variable concerned. For example, for a continuous random variable $X$, we write $p(x)$ to represent the density function. When necessary for discrete random variables, we write $p(S = j)$ to indicate the mass function of the discrete random variable $S$.

For the joint distribution of continuous random variable $X$ and discrete random variable $S$, we
write the mixed density and mass function

\[ p(x, S = j) = p(x \mid S = j)p(S = j) \]

or

\[ p(x, S = j) = p(S = j \mid x)p(x). \]

Hereafter, we will refer to mixed density and mass functions as simply density functions. For a random variable \( Y \) with density function \( p(y \mid \theta) \) and parameter(s) \( \theta \), we write the log-likelihood function

\[ l(\theta \mid y) = \log[p(y \mid \theta)]. \]

For a continuous random variable \( X \), we write the expectation

\[ E_X(X) = \int_{\Omega_X} x \, p(x) \, dx. \]

For a discrete random variable \( S \), we write the expectation

\[ E_S(S) = \sum_{s \in \Omega_S} s \, p(s). \]

**Definition 2.1 Independent Random Variables.** Random variables \( X \) and \( Y \) are independent, denoted \( X \perp \! \! \! \perp Y \), if and only if

\[ p(x, y) = p(x)p(y) \]

for all \( x \in \Omega_X \) and \( y \in \Omega_Y \).

An equivalent formulation of the independence property is

\[ X \perp \! \! \! \perp Y \iff p(x \mid y) = p(x) \]

for all \( x \in \Omega_X \) and \( y \in \Omega_Y \) such that \( p(y) > 0 \).
Definition 2.2 Conditionally Independent Random Variables. Random variables $X$ and $Y$ are conditionally independent given $Z$, denoted $X \perp \!\!\!\!\!\!\!\!\!\perp Y \mid Z$, if and only if

$$p(x, y|z) = p(x|z)p(y|z)$$

for all $x \in \Omega_X$, $y \in \Omega_Y$ and $z \in \Omega_Z$ such that $p(z) > 0$.

An equivalent formulation of the conditional independence property is

$$X \perp \!\!\!\!\!\!\!\!\!\perp Y \mid Z \iff p(x|y, z) = p(x|z)$$

for all $x \in \Omega_X$, $y \in \Omega_Y$ and $z \in \Omega_Z$ such that $p(y|z) > 0$ and $p(z) > 0$.

Theorem 2.1 Composition and Decomposition of Independence. For random variables $X_1, X_2$ and $Y$,

$$\{X_1, X_2\} \perp Y \iff X_1 \perp Y \text{ and } X_2 \perp Y \mid X_1.$$ 

Proof of Theorem 2.1. ($\Rightarrow$)

$$\{X_1, X_2\} \perp Y \iff p(x_1, x_2, y) = p(x_1, x_2)p(y)$$

$$\Rightarrow \int_{x_2 \in \Omega_{X_2}} p(x_1, x_2, y) \, dx_2 = \int_{x_2 \in \Omega_{X_2}} p(x_1, x_2)p(y) \, dx_2$$

$$\Rightarrow p(x_1, y) = p(x_1)p(y)$$

$$\iff X_1 \perp Y.$$ 

And

$$\{X_1, X_2\} \perp Y \Rightarrow p(x_1, x_2, y) = p(x_1, x_2)p(y)$$

$$= p(x_2|x_1)p(x_1)p(y)$$

$$= p(x_2|x_1)p(x_1, y)$$

$$\iff \frac{p(x_1, x_2, y)}{p(x_1, y)} = p(x_2|x_1)$$

$$\iff p(x_2|x_1, y) = p(x_2|x_1)$$

$$\iff X_2 \perp Y \mid X_1.$$ 

The third line follows as $X_1 \perp Y$ from above.
(⇐)

\[ p(x_1, x_2 | y) = p(x_2 | x_1, y) p(x_1 | y) \]
\[ = p(x_2 | x_1) p(x_1) \]
\[ = p(x_1, x_2). \]

The second line follows as \( X_1 \perp \perp Y \) and \( X_2 \perp \perp Y | X_1 \). Hence \( X_1 \perp \perp Y \) and \( X_2 \perp \perp Y | X_1 \). \( \square \)

**Corollary 2.2.** For random variables \( X_1, X_2, Y \) and \( Z \),

\[ \{X_1, X_2\} \perp \perp Y \mid Z \iff X_1 \perp \perp Y \mid Z \text{ and } X_2 \perp \perp Y \mid \{X_1, Z\}. \]

### 2.2 Sequences of Random Variables

**Definition 2.3 Discrete Time Stochastic Process.** A discrete time stochastic process is a sequence of random variables \( X_1, X_2, \ldots \) where the running index represents discrete time.

We will use set notation in the indices of random variables to denote collections of random variables. We always consider finite discrete time stochastic process of length \( T \) in order to use the following notation. Consider the discrete time stochastic process \( X_1, X_2, \ldots, X_T \). For any subset \( \mathcal{A} \subseteq \{1, 2, \ldots, T\} \), we write \( X_\mathcal{A} \equiv \{X_t \mid t \in \mathcal{A}\} \). Intersections, complements and other set operators appearing in the subscript are all well-defined in this way. We additionally write \( X_{1:t} = X_1, X_2, \ldots, X_t \) and \( X_{\infty:t} = X_1, \ldots, X_{t-1}, X_{t+1}, \ldots, X_T \) for \( t = 1, 2, \ldots, T \). Such notation will also be used for collections of realisations of these random variables.

**Definition 2.4 Discrete Time Markov Chain.** A discrete time Markov chain is a discrete time stochastic process \( S_{1:T} \) such that

\[ S_{1:t-1} \perp \perp S_{t+1} \mid S_t \]

for \( t = 2, 3, \ldots, T - 1 \).

We consider that the random variables that form a Markov chain are discrete valued and take values in a common state space \( \Omega_S = \{1, 2, \ldots, N\} \) so that \( s_t \in \Omega_S \) for \( t = 1, 2, \ldots, T \). We say that the Markov chain is ‘in state \( j \) at time \( t \)’ when \( S_t = j \) and that the Markov chain has ‘transitioned
from state \( i \) to state \( j \) in a single time step’ when \( S_{t-1} = i \) and \( S_t = j \).

**Definition 2.5 Time Homogeneous Markov Chain.** A discrete time Markov chain \( S_{1:T} \) is time homogenous if

\[
p(S_t = j|S_{t-1} = i) = p(S_2 = j|S_1 = i)
\]

for \( t = 2, 3, \ldots, T \) and \( i, j = 1, 2, \ldots, N \).

Hereafter, we always consider that \( S_{1:T} \) is a discrete time, time homogeneous Markov chain and that all of the random variables in the chain are discrete and take values in the common state space \( \Omega_S = \{1, 2, \ldots, N\} \).

### 2.3 Graphical Models

**Definition 2.6 Undirected Graph.** An undirected graph \( G \) is a set of vertices \( V \) and a set of edges \( E \), written \( G = (V, E) \) where \( E \) is a subset of the set \( V \times V \) of pairs of distinct vertices such that \( (u, v) \in E \Leftrightarrow (v, u) \in E \forall u, v \in V \).

**Definition 2.7 Paths and Separation.** For an undirected graph \( G = (V, E) \), a path between two vertices \( u, v \in V \) is a sequence \( w_1, w_2, \ldots, w_K \) such that \( w_1 = u, w_K = v \) and \( (w_{k-1}, w_k) \in E \) for \( k = 2, 3, \ldots, K \). Two vertices \( u, v \in V \) are separated by the subset \( C \subseteq V \setminus \{u, v\} \) if and only if all paths between \( u \) and \( v \) pass through at least one member of \( C \).

**Definition 2.8 Conditional Independence Graph.** The conditional independence graph of the collection of random variables \( X_V \) is the undirected graph \( G = (V, E) \) where

\[
(u, v) \notin E \Leftrightarrow X_u \perp \perp X_v \mid X_{V \setminus \{u, v\}}.
\]  

(2.1)

Collections of random variables with corresponding conditional independence graphs are also known as undirected graphical models or Markov random fields. They are a special case of Bayesian graphical models, also known as dynamic Bayesian networks (Bilmes 2006). For a review of dynamic Bayesian networks and preliminary discussion on using such models for time course microarray data, see the work of Murphy & Mian (1999). We use conditional independence graphs to visualise the conditional independence structure of collections of random variables. The content of this section is primarily based on the work of Lauritzen (1996) and Whittaker (1990).

Figure 2.1 is the conditional independence graph of a Markov chain \( S_{1:T} \) with vertices corresponding to the random variables \( S_{t-1}, S_t \) and \( S_{t+1} \). The following theorem allows for conditional
independence properties to be deduced from the conditional independence graph.

**Definition 2.9 Separation Condition.** The collection of random variables $X_V$ is said to satisfy the separation condition if for all disjoint subsets $A, B, C, D \subset V$,

$$ X_A \perp \perp X_B | X_{C \cup D} \quad \text{and} \quad X_A \perp \perp X_C | X_{B \cup D} \Rightarrow X_A \perp \perp X_{B \cup C} | X_D. \quad (2.2) $$

**Theorem 2.3 Separation Theorem.** If $X_A, X_B$ and $X_C$ are disjoint subsets of random variables from the collection $X_V$ that satisfies the separation condition and if in the conditional independence graph of $X_V$ each vertex in $A$ is separated from each vertex in $B$ by the subset $C$, then

$$ X_A \perp \perp X_B | X_C \quad (2.3) $$

**Proof of Theorem 2.3.** A proof is given by Lauritzen (1996). \(\square\)

We have defined conditional independence graphs using the pairwise Markov property (2.1). Theorem 2.3 is stated in terms of the global Markov property (2.3), which implies the pairwise Markov property (Lauritzen 1996). As shown by Lauritzen (1996), the pairwise Markov property implies the global Markov property provided the collection $X_V$ satisfies the separation condition.

It is easy to see that a Markov chain $S_{1:T}$ satisfies the separation condition and hence Theorem 2.3 holds. In order for the left hand side of (2.2) to hold, $t \in \mathcal{D}$ and, either $A \subseteq \{1, 2, \ldots, t-1\} \setminus \mathcal{D}$ and $B, C \subseteq \{t+1, t+2, \ldots, T\} \setminus \mathcal{D}$, or $A \subseteq \{t+1, t+2, \ldots, T\} \setminus \mathcal{D}$ and $B, C \subseteq \{1, 2, \ldots, t-1\} \setminus \mathcal{D}$, in which case the right hand side of (2.2) also holds for $t = 2, 3, \ldots, T-1$. Hence we can use Theorem 2.3 to read off conditional independence properties from Figure 2.1. This is summarised in the following lemma.

**Lemma 2.4.** An equivalent definition of a Markov chain is a collection of random variables $S_{1:T}$ with conditional independence graph given in Figure 2.1.
Proof of Lemma 2.4. The result is a consequence of the following two statements:

1. By Definition 2.8, for the collection $S_{1:T}$ where (2.4) holds, the conditional independence graph is given in Figure 2.1; and

2. By Theorem 2.3, for the collection $S_{1:T}$ with conditional independence graph given in Figure 2.1, it follows that (2.4) holds.

The first statement can be verified by considering the construction of a conditional independence graph given by Definition 2.8. Recall that there is no edge between two vertices if the corresponding random variables are conditionally independent given the rest of the collection. It is easy to see that this is true in Figure 2.1 by considering each possible pairing of the random variables in a Markov chain and the conditional independence property (2.4). The second statement can be verified by considering that in Figure 2.1, any path from the vertices corresponding to $S_{t-1}$ to the vertex corresponding to $S_{t+1}$ must pass through the vertex corresponding to $S_t$. Hence we have (2.4) for $t = 2, 3, \ldots, T - 1$ by Theorem 2.3.
Chapter 3

Hidden Markov Models

Hidden Markov models (HMMs) have been used to model time series data in a wide range of applications. Since the 1980s, HMMs have been the predominant model in automatic speech recognition systems (Bilmes 2006) and have been used to model phenomena such as animal behaviour, weather systems and financial markets (MacDonald & Zucchini 1997). HMMs have also been used in the analysis of time course microarray data (Bar-Joseph 2004).

An HMM is composed of two discrete time stochastic processes that we call the emission sequence and the state sequence of an HMM. Generally, the emission sequence is observed while the corresponding realisation of the state sequence, which is a Markov chain (Section 3.1), is unobserved and hence, ‘hidden Markov model’. An HMM is an extension of a Markov model where the dynamics of the additional emission sequence are driven by the Markov process. In most applications, the underlying state sequence is of primary interest.

In the following, we define and describe HMMs, making particular use of conditional independence graphs (Section 3.1), and then give a parameterisation of the model (Section 3.2). The content of this chapter is based on the work of Bilmes (2006) as well as the HMM review paper by Rabiner (1989).

3.1 Definition

Following Bilmes (2006), we consider an HMM as a finite collection of random variables. The model is defined and solutions to the associated problems (Chapter 4) are made tractable by specifying the conditional independence properties of such a collection.
Definition 3.1 Hidden Markov Model. A hidden Markov model (HMM) is a collection of random variables \( \{X_{1:T}, S_{1:T}\} \) such that

\[
\{X_{t-1:T}, S_{t-1}\} \perp \perp \{X_{t+1:T}, S_{t+1:T}\} \mid S_t
\]

for \( t = 2, 3, \ldots, T - 1 \) and

\[
X_t \perp \perp \{X_{\sim t}, S_{\sim t}\} \mid S_t
\]

for \( t = 1, 2, \ldots, T \).

We call \( X_{1:T} \) the emission sequence and \( S_{1:T} \) the state sequence of an HMM. The two properties (3.1) and (3.2) can be understood as a general Markov property and a stronger conditional independence property of the emission random variables respectively.

The state sequence is a Markov chain as \( S_{t-1} \perp \perp S_{t+1} \mid S_t \) for \( t = 2, 3, \ldots, T - 1 \) from (3.1) by decomposition (Corollary 2.2)\(^1\). It is also the case that subsequent emission random variables are conditionally independent of previous emission random variables from (3.1), however given the state random variable at the intermediate time. Moreover from (3.2), an emission random variable is conditionally independent of all other random variables given the state random variable at the corresponding time. Both of these observations conform to the notion that the Markov chain state sequence is the driving process in the model.

Consider the conditional independence graph of an HMM given in Figure 3.1. It is easy to see that an HMM satisfies the separation condition (2.2) since the same condition is satisfied by a Markov chain and the additional conditional independence structure of the emission random variables is completely dependent on the Markov chain structure. Hence we can use Theorem

\(^1\)Note that although we will continually invoke the decomposition result of Corollary 2.2 we will usually not explicitly cite it.
to read off conditional independence properties from Figure 3.1 and we therefore obtain the analogous result as in Lemma 2.4. That is, an equivalent definition of an HMM is a collection of random variables \( \{X_{1:T}, S_{1:T}\} \) with conditional independence graph given in Figure 3.1.

### 3.2 Parameterisation

We parameterise HMMs by considering the joint density of the emission and state sequences. Write

\[
p(s_{1:T}) = p(s_T|s_{1:T-1})p(s_{1:T-1})
\]

\[
= p(s_{T}|s_{1:T-1})p(s_{T-1}|s_{1:T-2})p(s_{1:T-2})
\]

\[
\vdots
\]

\[
= p(s_1) \prod_{t=2}^{T} p(s_t|s_{1:t-1}). \tag{3.3}
\]

Then the joint density of the emission and state sequences is

\[
p(x_{1:T}, s_{1:T}) = p(s_{1:T})p(x_{1:T}|s_{1:T})
\]

\[
= p(s_1) \prod_{t=2}^{T} p(s_t|s_{1:t-1}) \left( \prod_{t=1}^{T} p(x_t|s_{1:T}) \right)
\]

\[
= p(s_1) \prod_{t=2}^{T} p(s_t|s_{1:t-1}) \left( \prod_{t=1}^{T} p(x_t|s_t) \right). \tag{3.4}
\]

The second line follows from (3.3) and as \( X_t \perp X_{<t} \mid S_t \) for \( t = 1, 2, \ldots, T \) from (3.2). The third line follows as \( S_{1:t-1} \perp S_{t+1} \mid S_t \) for \( t = 2, 3, \ldots, T-1 \) from (3.1) and as \( X_t \perp S_{<t} \mid S_t \) for \( t = 1, 2, \ldots, T \) from (3.2). We can clearly see that the HMM density is the product of the Markov chain component of the model and the conditional emission densities.

Recall that the Markov chain state sequence \( S_{1:T} \) is discrete and time homogeneous. We consider that

\[
X_t \mid \{S_t = j\} \sim N(\mu_j, \sigma_j^2)
\]

for \( t = 1, 2, \ldots, T \) and hence have the following parameterisation for an HMM.

**Definition 3.2 HMM Parameters.** Let \( a = (a_1, a_2, \ldots, a_N)^T \) be the \( N \times 1 \) vector of initial state probabilities and \( A = \{a_{ij}\} \) be the \( N \times N \) state transition matrix of the Markov chain state
sequence where

\[ a_i = p(S_1 = i) \]

for \( i = 1, 2, \ldots, N \) and

\[ a_{ij} = p(S_t = j | S_{t-1} = i) \]

for \( i, j = 1, 2, \ldots, N \).

Let \( B = \{\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \ldots, \mu_N, \sigma_N^2\} \) be the set of all parameters of the emission distributions so that

\[ p(x_i | S_t = j) = b(x_i | \mu_j, \sigma_j^2) \]

where

\[ b(x_i | \mu_j, \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left\{ -\frac{1}{2\sigma_j^2} (x_i - \mu_j)^2 \right\} \]

for \( j = 1, 2, \ldots, N \).

Write the set of HMM parameters

\[ \lambda \equiv \{a, A, B\}. \]

Hence from (3.4) and Definition 3.2 we have the density function

\[ p(x_{1:T}, s_{1:T} | \lambda) = a_{s_1} \prod_{t=2}^{T} a_{s_{t-1} s_t} \left( \prod_{t=1}^{T} b(x_t | \mu_{s_t}, \sigma_{s_t}^2) \right), \]

(3.5)
Chapter 4

Associated Problems for HMMs

There exist three canonical problems associated with HMMs, the solutions of which allow for the model to be used in practice. Recall that in practice, realised state sequences are ‘hidden’ and so we require the maximum likelihood estimator of the model parameters given the observed emission sequence only. As the state sequence is of primary interest, we then require subsequent use of the estimated model to find the most likely corresponding realisation of the state sequence given an observed emission sequence. These outcomes are attained from the solutions to the following problems:

1. **Evaluation of the Marginal Emission Density** (Section 4.1)
   
   Given an observed emission sequence of an HMM with known parameters \( \lambda \), evaluate the marginal emission density at these observations.

2. **Most Likely Corresponding Realised State Sequence** (Section 4.2)
   
   Given an observed emission sequence of an HMM with known parameters \( \lambda \), find the most likely corresponding realisation of the state sequence.

3. **Maximum Likelihood Estimator of the Parameters** (Section 4.3)
   
   Given the observed emission sequences of \( K \) independent and identically distributed HMMs, find the maximum likelihood estimator of the parameters \( \lambda \).

Mathematical solutions to these problems were developed in the late 1960s and early 1970s, and are outlined in particular in the HMM review paper by Rabiner (1989). Our presentation is also based on the solutions as presented by Bilmes (1998) and Bilmes (2006).
4.1 Evaluation of the Marginal Emission Density

Suppose we have observed the emission sequence of an HMM with known parameters \( \lambda \). We evaluate the marginal emission density at this observation. This can be achieved by marginalising the joint density of the emission and state sequences. However such a procedure is computationally infeasible since

\[
p(x_{1:T} | \lambda) = \sum_{s_1=1}^{N} \sum_{s_2=1}^{N} \cdots \sum_{s_T=1}^{N} p(x_{1:T}, s_1:T | \lambda)
= \sum_{s_1=1}^{N} \sum_{s_2=1}^{N} \cdots \sum_{s_T=1}^{N} a_{s_1} \prod_{t=2}^{T} a_{s_{t-1}s_t} \left( \prod_{t=1}^{T} b(x_t | \mu_{s_t}, \sigma_{s_t}^2) \right).
\] (4.1)

That is, for each possible realisation of the state sequence, we evaluate the joint density of the emission and state sequences, which we sum over to obtain the marginal emission density. This procedure requires in the order of \( N^T \) calculations and hence is computationally infeasible even for small values of \( N \) and \( T \). In the following we present the standard solution to this evaluation problem using the forward and backward functions.

**Definition 4.1 Forward and Backward Functions.** Define the forward function

\[
\alpha_t(j | \lambda) = p(x_{1:t}, S_t = j | \lambda)
\] (4.2)

and the backward function

\[
\beta_t(j | \lambda) = p(x_{t+1:T} | S_t = j, \lambda).
\] (4.3)

**Theorem 4.1 Forward-Backward Evaluation (Solution to the First Problem).** Let the collection of random variables \( \{X_{1:T}, S_{1:T}\} \) be an HMM with known parameters \( \lambda \). Given an observed emission sequence, the marginal emission density is

\[
p(x_{1:T} | \lambda) = \sum_{j=1}^{N} \alpha_t(j | \lambda) \beta_t(j | \lambda)
\] (4.4)

for \( t = 1, 2, \ldots, T \).

We present the standard proof of Theorem 4.1 and show that evaluating the marginal emission density in this way is considerably less computationally expensive than evaluating (4.1). To do so, we first consider lemmas regarding the evaluation of both the forward and backward functions.
Lemma 4.2 Forward Evaluation. For the forward function (4.2),

$$\alpha_1(i|\lambda) = b(x_1|\mu_i, \sigma_i^2)a_i$$  \hspace{1cm} (4.5)

and

$$\alpha_t(j|\lambda) = b(x_t|\mu_j, \sigma_j^2)\sum_{i=1}^{N} a_{ij}\alpha_{t-1}(i|\lambda)$$  \hspace{1cm} (4.6)

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

Proof of Lemma 4.2. For notational convenience we suppress the conditioning symbol $\lambda$ as the parameters are known and considered fixed.

$$\alpha_1(i) = p(x_1, S_1 = i)$$

$$= p(x_1|S_1 = i)p(S_1 = i)$$

$$= b(x_1|\mu_i, \sigma_i^2)a_i$$

and

$$\alpha_t(j) = p(x_{1:t}, S_t = j)$$

$$\quad = \sum_{i=1}^{N} p(x_{1:t}, S_{t-1} = i, S_t = j)$$

$$\quad = \sum_{i=1}^{N} p(x_{1:t}, S_t = j|x_{1:t-1}, S_{t-1} = i)p(x_{1:t-1}, S_{t-1} = i)$$

$$\quad = \sum_{i=1}^{N} p(x_{1:t-1}, S_{t-1} = i, S_t = j)p(S_t = j|x_{1:t-1}, S_{t-1} = i)p(x_{1:t-1}, S_{t-1} = i)$$

$$\quad = p(x_t|S_t = j)\sum_{i=1}^{N} p(S_t = j|S_{t-1} = i)p(x_{1:t-1}, S_{t-1} = i)$$

$$\quad = b(x_t|\mu_j, \sigma_j^2)\sum_{i=1}^{N} a_{ij}\alpha_{t-1}(i).$$

The fifth line follows as $X_t \perp \{X_{1:t-1}, S_{t-1}\} \mid S_t$ from (3.2) and as $X_{1:t-1} \perp \perp S_t \mid S_{t-1}$, which can be seen from Figure 3.1 using Theorem 2.3. \qed

We consider the evaluation of the forward and backward functions using trellis diagrams (Rabiner 1989). In Figure 4.1, each vertex of the trellis corresponds to a value of the forward
function (4.2) for $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N$. Directed edges between vertices indicate the contribution to, and order in which the vertices are evaluated, only shown here for the vertex corresponding to $\alpha_2(1)$. For the forward function, the trellis is evaluated by moving forward through time, completely evaluating the vertices corresponding to each time point in turn.

![Trellis diagram of the forward function.](image)

To evaluate the first time segment of the trellis, we need to evaluate $\alpha_1(i)$ for $i = 1, 2, \ldots, N$, which requires $N$ multiplications from (4.5). Now consider evaluating $\alpha_2(1)$. This requires $N + 1$ multiplications from (4.6) and such a procedure needs to be carried out for each $\alpha_2(j)$ for $j = 1, 2, \ldots, N$ resulting in $N(N + 1)$ multiplications to evaluate the second time segment. Then the third time segment is similarly evaluated in $N(N + 1)$ multiplications, and so on through time. Hence the total number of multiplications required to completely evaluate the trellis is $N + N(N + 1)(T - 1)$ and the total number of additions required can be seen to be $N(N - 1)(T - 1)$.

**Lemma 4.3 Backward Evaluation.** For the backward function (4.3),

$$\beta_T(j|\lambda) = 1$$
for \( j = 1, 2, \ldots, N \) and

\[
\beta_{t-1}(i|\lambda) = \sum_{j=1}^{N} \beta_t(j|\lambda)b(x_t|\mu_j, \sigma_j^2)a_{ij}
\]

for \( t = 2, 3, \ldots, T \).

Proof of Lemma 4.3. As before, for notational convenience we suppress the conditioning symbol \( \lambda \).

By definition of the backward function, \( \beta_T(j) = 1 \) for \( j = 1, 2, \ldots, N \) and

\[
\beta_{t-1}(i) = p(x_t:T|S_{t-1} = i)
\]

\[
= \sum_{j=1}^{N} p(x_t:T, S_t = j|S_{t-1} = i)
\]

\[
= \sum_{j=1}^{N} p(x_{t+1:T}|x_t, S_{t-1} = i, S_t = j)p(x_t, S_t = j|S_{t-1} = i)
\]

\[
= \sum_{j=1}^{N} p(x_{t+1:T}|x_t, S_{t-1} = i, S_t = j)p(x_t|S_{t-1} = i, S_t = j)p(S_t = j|S_{t-1} = i)
\]

\[
= \sum_{j=1}^{N} \beta_t(j)p(x_t|S_t = j)p(S_t = j|S_{t-1} = i)
\]

\[
= \sum_{j=1}^{N} \beta_t(j)b(x_t|\mu_j, \sigma_j^2)a_{ij}.
\]

The fifth line follows as \( \{X_t, S_{t-1}\} \perp \perp X_{t+1:T} \mid S_t \), which can be seen from Figure 3.1 using Theorem 2.3, and as \( X_t \perp \perp X_{t+1:T} \mid S_t \) from (3.2).

We consider an analogous trellis diagram for the backward function but where the evaluation procedure is now required to move backward through time (Figure 4.2). It can be seen that to completely evaluate all of the vertices of the trellis in Figure 4.2 requires \( 2N^2(T - 1) \) multiplications and \( N(N - 1)(T - 1) \) additions. Hence we now prove Theorem 4.1 and validate the decrease in computational expense.
Figure 4.2: Trellis diagram of the backward function.

Proof of Theorem 4.1.

\[
p(x_{1:T} | \lambda) = \sum_{j=1}^{N} p(x_{1:T}, S_t = j | \lambda)
= \sum_{j=1}^{N} p(x_{1:t}, S_t = j | \lambda)p(x_{t+1:T} | x_{1:t}, S_t = j, \lambda)
= \sum_{j=1}^{N} p(x_{1:t}, S_t = j | \lambda)p(x_{t+1:T} | S_t = j, \lambda)
= \sum_{j=1}^{N} \alpha_t(j | \lambda) \beta_t(j | \lambda).
\]

The third line follows as \(X_{1:t} \perp X_{t+1:T} \mid S_t\), which can be seen from Figure 3.1 using Theorem 2.3.

To evaluate the marginal emission density using Theorem 4.1 requires the evaluation of the forward trellis up to time \(t\) and the backward trellis (back) to time \(t\). This requires a total of \(N + N(N + 1)(t - 1) + 2N^2(T - t)\) multiplications and \(N(N - 1)(t - 1) + N(N - 1)(T - t)\) additions. We then require an additional \(N\) multiplications and \(N - 1\) additions to evaluate (4.4). This procedure is a considerable decrease in computational expense compared to evaluating the marginal density using (4.1), which required in the order of \(N^T\) calculations.
4.2 Most Likely Corresponding Realised State Sequence

Suppose we have observed the emission sequence of an HMM with known parameters $\lambda$. We find the most likely corresponding realisation of the state sequence. The overall most likely such sequence is

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}} p(s_{1:T}|x_{1:T}),$$

where $x_{1:T}$ is the observed emission sequence and considered fixed. Note that we suppress the conditioning symbol $\lambda$ in any densities as the parameters are known and considered fixed. Since

$$p(s_{1:T}|x_{1:T}) = \frac{p(x_{1:T}, s_{1:T})}{p(x_{1:T})},$$

maximising $p(s_{1:T}|x_{1:T})$ with respect to $s_{1:T}$ is the same as maximising $p(x_{1:T}, s_{1:T})$ with respect to $s_{1:T}$ and hence we equivalently have

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}} p(x_{1:T}, s_{1:T}). \tag{4.7}$$

Finding $\hat{s}_{1:T}$ could be achieved by calculating $p(x_{1:T}, s_{1:T})$ for each possible realisation of the state sequence and taking $\hat{s}_{1:T} = s_{1:T}$ that maximises the joint density. However, we then have the same issue as for the first problem (Section 4.1), namely that such a procedure is computationally infeasible. We present the standard solution to this problem using the Viterbi function.

**Definition 4.2 Viterbi Function.** Define the Viterbi function

$$v_t(j) = \max_{s_{t-1}} p(x_{1:t}, s_{1:t-1}, S_t = j). \tag{4.8}$$

**Theorem 4.4 Viterbi Path (Solution to the Second Problem).** Let the collection of random variables $\{X_{1:T}, S_{1:T}\}$ be an HMM with known parameters $\lambda$. Given an observed emission sequence, the overall most likely corresponding realisation of the state sequence is $\hat{s}_{1:T}$ where

$$\hat{s}_{1:T} = \arg\max_{\hat{s}_{1:T}} [v_T(j)].$$
\[
\hat{s}_{t-1} = \arg\max_i [v_{t-1}(i) a_{i\hat{s}_t}]
\]

for \(t = 2, 3, \ldots, T\).

We call \(\hat{s}_{1:T}\) the Viterbi path corresponding to the observed emission sequence. This is an instance of a dynamical programming problem as each element of the most likely corresponding realisation of the state sequence must also be the most likely realisation up to that time point. Viterbi (1967) contributed to the theory that the Viterbi function and Viterbi path are based on.

We show that calculating the Viterbi path using Theorem 4.4 is much less computationally expensive than calculating the joint density for each possible realisation of the state sequence. We first consider the evaluation of the Viterbi function.

**Lemma 4.5 Viterbi Evaluation.** For the Viterbi function (4.8),

\[
v_1(i) = b(x_1|\mu_i, \sigma_i^2) a_i
\]

and

\[
v_t(j) = \max_i [v_{t-1}(i) a_{ij}] b(x_t|\mu_j, \sigma_j^2)
\]

for \(t = 2, 3, \ldots, T\).

**Proof of Lemma 4.5.**

\[
v_1(i) = p(x_1, S_1 = i)
\]

\[
= p(x_1|S_1 = i)p(S_1 = i)
\]

\[
= b(x_1|\mu_i, \sigma_i^2) a_i,
\]
and specified a running index over the possible realisations of \( S \).

The third line follows as \( X \) account all previous realisations and moving to the next known realised value \( \hat{s}_t \) ever, as we now know the subsequent realisation of the path, we have \( \hat{s}_T = \arg\max \) by definition of the Viterbi path (4.7) and the Viterbi function (4.8). It is

Proof of Theorem 4.4.

The third line follows as \( X_t \perp X_{1:t-1} \mid S_t \) from (3.2). The fourth line follows as \( X_{1:t-1} \perp S_t \mid S_{t-1} \), seen from Figure 3.1 using Theorem 2.3, and as \( X_t \perp S_{1:t-1} \mid S_t \) from (3.2). The sixth line follows as \( S_{1:t-2} \perp S_t \mid S_{t-1} \) from (3.1). The seventh line follows as we have broken up the max operator and specified a running index over the possible realisations of \( S_{t-1} \).

Evaluating the Viterbi function is essentially the same as for the forward function (Lemma 4.2) except we maximise rather than sum over a collection of quantities corresponding to the previous time point. Hence we obtain a similar trellis diagram for the Viterbi function as in Figure 4.1 and to completely evaluate (4.8) for all \( t = 1, 2, \ldots, T \) and \( j = 1, 2, \ldots, N \) can be seen to require \( N + N(N + 1)(T - 1) \) multiplications and \( N(T - 1) \) max operations over \( N \) quantities.

Proof of Theorem 4.4.

\[ \hat{s}_T = \arg\max_j [v_T(j)] \]

by definition of the Viterbi path (4.7) and the Viterbi function (4.8). It is not the case that \( \hat{s}_{T-1} = \arg\max_i [v_{T-1}(i)] \) as this does not take into account the subsequent realisation of the path. However, as we now know the subsequent realisation of the path, we have \( \hat{s}_{T-1} = \arg\max_i [v_{T-1}(i)a_{ij}] \).

That is, \( \hat{s}_{T-1} \) is the realisation that has maximum probability occurring at time \( T - 1 \) taking into account all previous realisations and moving to the next known realised value \( \hat{s}_T \). The same
CHAPTER 4. ASSOCIATED PROBLEMS FOR HMMS

argument applies for all preceding time points resulting in

$$\hat{s}_{t-1} = \text{argmax}_i [v_{t-1}(i)a_{i\hat{s}_t}]$$

for $t = 2, 3, \ldots, T$. \hfill \square

Hence, given the Viterbi function (4.8) evaluated for all $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N$, the procedure for obtaining the Viterbi path requires $N(T - 1)$ multiplications and $T \text{ argmax}$ operations over $N$ quantities. This is a considerable decrease in computational expense compared to evaluating the joint density for each possible realisation of the state sequence, which required in the order of $N^T$ calculations.

4.3 Maximum Likelihood Estimator of the Parameters

Suppose we have observed the emission sequences of $K$ independent and identically distributed HMMs. We find the maximum likelihood estimator (MLE) of the parameters $\lambda$. Write the $k^{th}$ emission sequence

$$X_{1:T}^{(k)} = X_1^{(k)}, X_2^{(k)}, \ldots, X_T^{(k)}$$

and the collection of $K$ emission sequences

$$X_{1:T}^{(1:K)} = \{X_{1:T}^{(1)}, X_{1:T}^{(2)}, \ldots, X_{1:T}^{(K)}\}.$$ 

Write the collection of $K - 1$ emission sequences excluding the $k^{th}$ sequence

$$X_{1:T}^{(\sim k)} = \{X_{1:T}^{(1)}, \ldots, X_{1:T}^{(k-1)}, X_{1:T}^{(k+1)}, \ldots, X_{1:T}^{(K)}\}$$

for $k = 1, 2, \ldots, K$. We use the same notation for the collection of $K$ state sequences and for realisations of these random variables. Hence the independence assumption is

$$\{X_{1:T}^{(k)}, S_{1:T}^{(k)}\} \perp \perp \{X_{1:T}^{(\sim k)}, S_{1:T}^{(\sim k)}\}, \quad (4.9)$$

where each collection $\{X_{1:T}^{(k)}, S_{1:T}^{(k)}\}$ is an HMM with unknown parameters $\lambda$ for $k = 1, 2, \ldots, K$. 
The maximum likelihood estimator is
\[ \hat{\lambda} = \arg\max_{\lambda} l(\lambda|x^{(1:K)}_{1:T}). \]

No analytic solution exists if we wish to maximise \( l(\lambda|x^{(1:K)}_{1:T}) \) with respect to \( \lambda \). We have the joint density
\[
p(x^{(1:K)}_{1:T}, s^{(1:K)}_{1:T} | \lambda) = \prod_{k=1}^{K} \left[ \sum_{s_{1}^{(k)}} \sum_{s_{2}^{(k)}} \cdots \sum_{s_{T}^{(k)}} a_{s_{1}^{(k)}} \prod_{t=2}^{T} a_{s_{t-1}^{(k)} s_{t}^{(k)}} \left( \prod_{t=1}^{T} b(x_{t}^{(k)} | \mu_{s_{t}^{(k)}}, \sigma_{s_{t}^{(k)}}^{2}) \right) \right].
\]

and the marginal emission density
\[
p(x^{(1:K)}_{1:T} | \lambda) = \prod_{k=1}^{K} \left[ \sum_{s_{1}^{(k)}} \sum_{s_{2}^{(k)}} \cdots \sum_{s_{T}^{(k)}} a_{s_{1}^{(k)}} \prod_{t=2}^{T} a_{s_{t-1}^{(k)} s_{t}^{(k)}} \left( \prod_{t=1}^{T} b(x_{t}^{(k)} | \mu_{s_{t}^{(k)}}, \sigma_{s_{t}^{(k)}}^{2}) \right) \right].
\]

Hence
\[
l(\lambda|x^{(1:K)}_{1:T}) = \log \prod_{k=1}^{K} \left[ \sum_{s_{1}^{(k)}} \sum_{s_{2}^{(k)}} \cdots \sum_{s_{T}^{(k)}} a_{s_{1}^{(k)}} \prod_{t=2}^{T} a_{s_{t-1}^{(k)} s_{t}^{(k)}} \left( \prod_{t=1}^{T} b(x_{t}^{(k)} | \mu_{s_{t}^{(k)}}, \sigma_{s_{t}^{(k)}}^{2}) \right) \right].
\]

So to obtain an analytic expression of the MLE requires maximising over logarithms of sums of the parameters, which is intractable. The standard solution to this problem is given by the Baum-Welch algorithm (Rabiner 1989).

Algorithm 4.1 Baum-Welch Algorithm (Solution to the Third Problem). Let the collection of random variables \( \{X_{1:T}^{(k)}, S_{1:T}^{(k)}\} \) be an independent and identically distributed HMM for \( k = 1, 2, \ldots, K \). Given observed emission sequences, to obtain the maximum likelihood estimate of the parameters \( \lambda \):

1. Obtain initial estimate of the parameters, \( \lambda^{[0]} \).
2. At the \( n^{th} \) iteration, calculate
\[
a^{[n]}_{i} = \frac{1}{K} \sum_{k=1}^{K} \zeta_{i}^{(k)}(i | \lambda^{[n-1]}),
\]
\[ a_{ij}^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=2}^{T} \zeta_{t-1}^{(k)}(i,j|\lambda^{[n-1]})}{\sum_{k=1}^{K} \sum_{t=2}^{T} \zeta_{t-1}^{(k)}(i|\lambda^{[n-1]})} \]  

(4.11)

\[ \mu_j^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta_{t}^{(k)}(j|\lambda^{[n-1]})x_t^{(k)}}{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta_{t}^{(k)}(j|\lambda^{[n-1]})} \]  

(4.12)

and

\[ (\sigma_j^2)^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta_{t}^{(k)}(j|\lambda^{[n-1]}) (x_t^{(k)} - \mu_j^{[n]})^2}{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta_{t}^{(k)}(j|\lambda^{[n-1]})} \]  

(4.13)

for \( i, j = 1, 2, \ldots, N \) where

\[ \zeta_{t}^{(k)}(j|\lambda^{[n-1]}) = p(S_t^{(k)} = j|x_{1:T}^{(k)}, \lambda^{[n-1]}) \]  

(4.14)

and

\[ \zeta_{t-1}^{(k)}(i,j|\lambda^{[n-1]}) = p(S_{t-1}^{(k)} = i, S_t^{(k)} = j|x_{1:T}^{(k)}, \lambda^{[n-1]}) \]  

(4.15)

3. Return to Step 2 for a specified number of iterations or until some convergence criterion is satisfied.

The Baum-Welch algorithm is a special case of the EM algorithm, used for finding the MLE of model parameters where there is incomplete or missing data (Appendix B). In this case the ‘hidden’ realised state sequences are the missing data. The EM algorithm as applied to the HMM MLE problem is known as the Baum-Welch algorithm as it was proposed prior to the general EM form by Baum, Petrie, Soules & Weiss (1970). In the following, we work through applying the EM algorithm to the HMM MLE problem in order to illustrate that the Baum-Welch algorithm is indeed a special case of the EM algorithm. We have the full data log-likelihood

\[ l(\lambda|\mathbf{x}_{1:T}, \mathbf{s}_{1:T}^{(1:K)}) = \log \left[ \prod_{k=1}^{K} a_{s_1^{(k)}} \prod_{t=2}^{T} \prod_{s_{t-1}^{(k)}=s_1^{(k)}} \left( \prod_{t=1}^{T} b(x_t^{(k)}|\mu_{s_t^{(k)}}, \sigma_{s_t^{(k)}}^2) \right) \right] \]

\[ = \sum_{k=1}^{K} \log[a_{s_1^{(k)}}] + \sum_{k=1}^{K} \sum_{t=2}^{T} \log[a_{s_{t-1}^{(k)}}] + \sum_{k=1}^{K} \sum_{t=1}^{T} \log[b(x_t^{(k)}|\mu_{s_t^{(k)}}, \sigma_{s_t^{(k)}}^2)]. \]
CHAPTER 4. ASSOCIATED PROBLEMS FOR HMMS

The EM algorithm (Algorithm B.1) requires that at the \( n \)th iteration we find

\[
\lambda^{[n]} = \arg \max_{\lambda} Q(\lambda|\lambda^{[n-1]})
\]

where the objective function is

\[
Q(\lambda|\lambda^{[n-1]}) = E_{S_t^{(1:T)}} \left[ l(\lambda|x_{1:T}^{(1:K)}, S_t^{(1:K)}) \left| x_{1:T}^{(1:K)}, \lambda^{[n-1]} \right. \right]. \tag{4.16}
\]

**Theorem 4.6 Analytic Parameter Updates.** Given current estimate \( \lambda^{[n-1]} \), the objective function (4.16) is locally maximised at \( \lambda^{[n]} \) given by (4.10), (4.11), (4.12) and (4.13).

**Proof of Theorem 4.6.** A proof is given in Appendix C.

Hence Step 2 of the Baum-Welch algorithm is both the Expectation and the Maximisation steps of the equivalent EM form. The following lemma on the evaluation of the weights (4.14) and (4.15) allows for the Baum-Welch algorithm to be used in practice.

**Lemma 4.7 Evaluation of Weights.** For the weights \( \zeta_t^{(k)}(j|\lambda^{[n-1]}) \) and \( \xi_t^{(k)}(i,j|\lambda^{[n-1]}) \) given by (4.14) and (4.15) respectively,

\[
\zeta_t^{(k)}(j|\lambda^{[n-1]}) = \frac{\alpha_t^{(k)}(j|\lambda^{[n-1]})\beta_t^{(k)}(j|\lambda^{[n-1]})}{\sum_{j=1}^{N} \alpha_t^{(k)}(j|\lambda^{[n-1]})\beta_t^{(k)}(j|\lambda^{[n-1]})}
\]

and

\[
\xi_t^{(k)}(i,j|\lambda^{[n-1]}) = \frac{\alpha_{t-1}^{(k)}(i|\lambda^{[n-1]})b(x_t^{(k)}|\mu_j, \sigma_j^2)\beta_t^{(k)}(j|\lambda^{[n-1]})a_{ij}}{\sum_{j=1}^{N} \alpha_{t-1}^{(k)}(i|\lambda^{[n-1]})b(x_t^{(k)}|\mu_j, \sigma_j^2)\beta_t^{(k)}(j|\lambda^{[n-1]})}
\]

where

\[
\alpha_t^{(k)}(j|\lambda^{[n-1]}) = p(x_{1:t}^{(k)}, S_t^{(k)} = j|\lambda^{[n-1]}) \tag{4.17}
\]

and

\[
\beta_t^{(k)}(j|\lambda^{[n-1]}) = p(x_{t+1:T}^{(k)}|S_t^{(k)} = j, \lambda^{[n-1]}). \tag{4.18}
\]

**Proof of Lemma 4.7.** Note that (4.17) and (4.18) are analogous forward and backward functions to those in Section 4.1, however specified for the collection \( \{X_{1:T}^{(k)}, S_{1:T}^{(k)}\} \). As each collection
The second line follows as \( X_{t} \perp \perp X_{t+1} \| S_{t} \) from (3.2) and then \( X_{1:t-1} \perp \perp X_{1:t+1} \| S_{t} \) from (3.1). The third line follows as \( X_{1:t-1} \perp \perp S_{1:t-1} \) from Figure 3.1 using Theorem 2.3, as \( X_{t} \perp \perp S_{t-1} \| S_{t} \) from (3.2) and as \( S_{t-1} \perp \perp X_{t+1:t} \| S_{t} \) from (3.1). Hence

\[
\xi_t(i, j) = \frac{p(x_{1:T}, S_{t-1} = i, S_t = j)}{p(x_{1:T})} = \frac{\alpha_{t-1}(i)b(x_t|\mu_j, \sigma^2_j)\beta_t(j)a_{ij}}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}.
\]
Chapter 5

Alignment with HMMs

We review a number of papers in which the authors proposed HMM based models for time course microarray data or in which the authors proposed HMM based methodologies for particular time series alignment problems. Although non-HMM alignment methodologies have been proposed for time course microarray data (Aach & Church 2001), most methods do not lead to obvious common representations of the aligned profiles (Tsiporkova & Boeva 2008). We focus exclusively on HMM based methods for this project.

We first consider the alignment of genomic sequences data, an area in which HMMs have been extensively used. We define and describe Pair HMMs (Section 5.1), the standard model in this area. We conclude that Pair HMMs cannot readily be applied to our problem and consider two broad ways in which Pair HMMs could be extended in order to model time course microarray data (Section 5.2). We review papers in which the authors propose models that we identify as one of these extensions (Section 5.2) and review additional papers in which the authors achieve an alignment using HMMs that cannot be considered as such an extension (Section 5.3).

5.1 Pair HMMs for Alignment of Genomic Sequences

Durbin et al. (1998) presented a review of the standard methods for the alignment of genomic sequence data using HMMs, the most basic methodology based on Pair HMMs. We present the typical problem considered by Durbin et al. (1998) and argue that although there are superficial similarities to our own problem, there are also fundamental differences. Hence we only present Pair HMMs in enough detail to demonstrate these similarities and differences. The specifics of the
model, such as the associated algorithms, were presented by Durbin et al. (1998) and can be found in their work.

The genomic sequences of interest are sections of a genome represented by sequences of letters corresponding to the four nucleotides, Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). We consider that such genomic sequences are situated in discrete time corresponding to the order of the sequence. The question of interest is to what extent two genomic sequences are related by considering how many elements of the sequences match at corresponding time points. Preserving the order of the genomic sequences, the aim is to match subsequent elements of the sequences by adding insert elements (_\_) into either sequence to change the positions of the elements at each time point and hence align the two sequences.

To begin, we present an example result to illustrate the problem. Consider aligning the two genomic sequences AATCGAGCTA and AACTGCGTA. By adding inserts in the sequences as shown in Figure 5.1, we obtain a total of 7 elements matching between the aligned output sequences. Durbin et al. (1998) achieved such alignment results by using a Pair HMM.

\[
\text{Input:} \quad \text{Output:}
\]

\[
\begin{array}{c}
\text{AATCGAGCTA} \\
\text{AACTGCGTA}
\end{array}
\begin{array}{c}
\text{AA_T_CGAGCTA} \\
\text{AACTGCG___TA}
\end{array}
\]

Figure 5.1: Schematic diagram of the alignment of two genomic sequences.

**Definition 5.1 Pair HMM.** A Pair HMM is a hidden Markov model \(\{X_{1:T}, S_{1:T}\}\) such that

\[
\Omega_S = \{\text{Pair}', \text{Insert 1}', \text{Insert 2}'\} \tag{5.1}
\]

and

\[
x_t \mid \{S_t = j\} \in \begin{cases} 
\{(m,n) \mid m, n \in \{A, C, G, T\}\} & \text{if } j = \text{Pair}' \\
\{(_-,n) \mid n \in \{A, C, G, T\}\} & \text{if } j = \text{Insert 1}' \\
\{(m,_-) \mid m \in \{A, C, G, T\}\} & \text{if } j = \text{Insert 2}'
\end{cases} \tag{5.2}
\]

for \(t = 1, 2, \ldots, T\).

From (5.1) we see that there are \(N = 3\) states, labelled ‘Pair’, ‘Insert 1’ and ‘Insert 2’. The
model is known as a Pair HMM as the emission random variables are realised as an ordered pair of nucleotide symbols or a nucleotide symbol paired with an insert symbol. For the emission distribution when $S_t = \text{‘Pair’}$, the realisations where $m = n$ are generally set to be much more likely than where $m \neq n$, which is why the ‘Pair’ state is usually denoted ‘Match’ in the literature (Yoon 2009). Note that for a Pair HMM, the realisation of the state sequence, which is generally considered ‘hidden’ for HMMs, is entirely identifiable through the corresponding observed emission sequence as can be seen from (5.2). In practice, the positions of the insert symbols in the observed emission sequences are ‘hidden’ as described below.

The pair of genomic sequences to be aligned are modelled as the two sequences made up of the first and second elements of the emission random variables respectively. The output sequence in Figure 5.1 is an example of an observed emission sequence from a Pair HMM where the first element of each emission observation is printed on top of the second. The pair of corresponding genomic sequences, the input in Figure 5.1, can be seen to be the ordered elements of the observed emission sequence without the inserts.

The particular novelty of the Pair HMM is that there exist algorithms to find the most likely corresponding realisation of the state sequence while simultaneously adding inserts to the input genomic sequences in order to additionally obtain the observed emission sequence. Recall that finding the most likely corresponding realised state sequence given an observed emission sequence is the second associated problem for HMMs (Section 4.2). Durbin et al. (1998) reviewed the algorithms used to simultaneously obtain both the most likely realised state sequence and the observed emission sequence, and hence the alignment of the genomic sequences.

Although Pair HMMs have been successfully used for particular alignment problems for genomic sequence data, they are not adequate for our purposes. The primary issue is that a Pair HMM requires discrete emission random variables to model the genomic sequences of interest. In order to model the expression profiles of the grapevine data, we require continuous emission random variables. There have been many extensions proposed for Pair HMMs and related models for genomic sequence data (Durbin et al. 1998, Yoon 2009), however the emission random variables are always discrete.

In addition, for a Pair HMM, the value of the state random variable indicates whether or not the emission observation is a pair of nucleotide symbols or a single nucleotide symbol paired with an insert. That is, the conditional information of a previous emission observation is not the actual observed value but whether the observation was a pair or single nucleotide symbol. We aim to
interpret the underlying Markov structure as capturing distinct quantitative levels of the expression profiles in our model for the grapevine data. Hence we require more than the binary pair/single nucleotide symbol dynamics of the Markov chain component of the Pair HMM.

5.2 Extensions of Pair HMMs

Two ways in which Pair HMMs could be extended to model time course microarray data are to:

1. Retain the binary dynamics of the Markov chain component of the model and consider continuous emission random variables; or

2. Incorporate additional information into the model so that the Markov structure encodes more than just binary dynamics.

Note that these possible extensions do not explicitly take alignment into account, although the motivation in considering such extensions is that the established alignment methodology of Pair HMMs could be carried over. Yuan & Kendziorski (2006) and Yoneya & Mamitsuka (2007) both proposed models that can be considered as an extension of a Pair HMM that retains the binary dynamics of the Markov chain component of the model. Both modelled time course microarray data and hence considered continuous emission random variables. Listgarten, Neal, Roweis & Emili (2004) proposed a model that can be seen as an extension of a Pair HMM where there has been additional information incorporated into the model.

5.2.1 Binary Markov Dynamics with Continuous Emissions

Yuan & Kendziorski (2006) aimed to identify equivalent or differential expression at each time point of equal length expression profiles corresponding to the same gene collected from multiple ‘biological conditions’. We present their proposed HMM for two biological conditions, for which they considered two types of states in their model, ‘equivalent expression’ (EE) and ‘differential expression’ (DE). Note that there may be more than two states in the model, but each state is labeled as either an EE state or a DE state.

In their proposed HMM, the emission random variables at each time point are random vectors with independent elements. The multiple elements constitute multiple emission sequences in order to model the multiple expression profiles corresponding to the same gene from different biological conditions. In our presentation of their model we consider that there are two biological conditions.
and a single expression profile from each condition for each gene. Write the emission random vector at time $t$ as

$$X_t = [X_t^{(1)}, X_t^{(2)}]^T. \quad (5.3)$$

Define the emission sequences $X_t^{(1)}$ and $X_t^{(2)}$ that are used to model the expression profiles from biological condition 1 and 2 respectively. Yuan & Kendziorski (2006) set

$$X_t^{(l)} | \{S_t = j\} \sim \Gamma(\alpha_j^{(l)}, \beta_j^{(l)})$$

where $\alpha_j^{(l)}$ and $\beta_j^{(l)}$ are the shape and rate parameters of the Gamma distribution respectively. If $j$ is a EE state then $\alpha_j^{(1)} = \alpha_j^{(2)}$ and $\beta_j^{(1)} = \beta_j^{(2)}$. That is, the emission distributions are identical for both biological conditions. If $j$ is a DE state, then there is no relationship between the different sets of distribution parameters and the two distributions are distinct.

Yuan & Kendziorski (2006) determined whether a gene is equivalently or differentially expressed at a single time point by considering the most likely corresponding realisation of the state random variable at that time, similar to finding the Viterbi path (Section 4.2). If the most likely realisation of the state random variable is an EE state then the gene is considered to be equivalently expressed at that time, similarly for a DE state. Yuan & Kendziorski (2006) did not consider whether a gene is equivalently or differentially expressed overall, they only addressed this question at individual time points.

Yoneya & Mamitsuka (2007) aimed at identifying the overall timing differences in gene expression profiles from different ‘experimental factors’. They assume that the profiles have close to average expression level at most time points with at least one spike of non-average expression. The different times at which these spikes of expression occur in the profiles corresponding to the same gene are the timing differences between different experimental factors. Yoneya & Mamitsuka (2007) modelled these timing differences using a proposed methodology based on HMMs.

Two types of states are proposed by Yoneya & Mamitsuka (2007), ‘control’ states and ‘feature’ states. The Markov state transition matrix is constrained so that the feature states may only be visited once and are visited in a prescribed order. In the same way as in the HMM proposed by Yuan & Kendziorski (2006), emission random vectors (5.3) with independent elements constitute multiple emission sequences to model the multiple expression profiles corresponding to the same

\[ X_t^{(l)} \mid \{ S_t = j \} \sim N(\mu_j^{(l)}, (\sigma_j^{(l)})^2). \]

If \( j \) is a control state then the corresponding emission distribution parameters are not iteratively trained using the Baum-Welch algorithm (Section 4.3) but are estimated by

\[
\hat{\mu}_j^{(l)} = \frac{1}{T} \sum_{t=1}^{T} x_t^{(l)}
\]

and

\[
(\hat{\sigma}_j^{(l)})^2 = \frac{1}{T} \sum_{t=1}^{T} (x_t^{(l)} - \hat{\mu}_j^{(l)})^2,
\]

corresponding to the average expression level over all time points. If \( j \) is a feature state, corresponding to a spike in expression level, then the parameters are iteratively trained along with the Markov model parameters. Yoneya & Mamitsuka (2007) identified the ‘timing difference’ in the observed expression profiles over the different experimental factors by considering the time at which the feature states are visited. They are clear that they are not modelling the expression profiles explicitly, but rather modelling the timing differences between profiles from different experimental factors.

To achieve their aims, Yuan & Kendziorski (2006) and Yoneya & Mamitsuka (2007) both proposed models that have similarities to Pair HMMs. Yuan & Kendziorski (2006) did not aim to obtain an alignment between expression profiles, nor is there an obvious way to adapt their method to do so. Yoneya & Mamitsuka (2007) modelled the timing differences between expression profiles and so this could be used as the basis of an alignment methodology. However, they made strict assumptions about the shape of the expression profiles, assuming average expression levels except for at least one spike feature. We know that this assumption would generally not hold for the grapevine data and so their methodology would not be appropriate as a basis for alignment in this case.

### 5.2.2 Additional Information Incorporated into Model

Listgarten et al. (2004) were interested in aligning replicate time series associated with the same process in order to obtain a single representation of all the observed series. For this task, they proposed their Continuous Profile Model (CPM), which they consider to be a ‘continuous analogue’ to a Profile HMM. Very similar to Pair HMMs, Profile HMMs have also been extensively used for alignment problems in genomic sequences data (Durbin et al. 1998, Yoon 2009). Listgarten et al. (2004) additionally point out that their CPM has many similarities to Input/Output HMMs, extensions to HMMs that were described by Bengio & Frasconi (1995).
Definition 5.2 Input/Output HMM. An Input/Output HMM (IOHMM) is a collection of random variables \( \{X_{1:T}, S_{1:T}, Z_{1:T}\} \) where \( \{X_{1:T}, S_{1:T}\} \) is a hidden Markov model such that

\[
\{X_t, S_t\} \perp \perp Z_{1:t-1} \mid S_{t-1} \quad (5.4)
\]

for \( t = 2, 3, \ldots, T, \)

\[
\{X_t, S_t\} \perp \perp Z_{t+1:T} \quad (5.5)
\]

for \( t = 1, 2, \ldots, T - 1 \) and

\[
Z_t \perp \perp Z_{\sim t} \quad (5.6)
\]

for \( t = 1, 2, \ldots, T. \)

Additional information is incorporated into the IOHMM in the form of an additional input sequence \( Z_{1:T} \) that conditions both the emission and state sequences. By considering the probability density function of an HMM (Section 3.2) it is easy to see from (5.4) and (5.5) that

\[
p(x_{1:T}, s_{1:T}, z_{1:T}) = p(z_{1:T})p(s_1 | z_1) \prod_{t=2}^{T} p(s_t | s_{t-1}, z_t) \left( \prod_{t=1}^{T} p(x_t | s_t, z_t) \right).
\]

Note that there is no alignment under an IOHMM, nor are IOHMMs extensions of Pair HMMs, nor is there an equivalent conditional independence graph for IOHMMs due to the marginal independence conditions (5.5) and (5.6). Recall that Listgarten et al. (2004) considered their CPM to be a ‘continuous analogue’ to a Profile HMM but also that their CPM has many similarities to IOHMMs. We have presented the definition of IOHMMs to give the basic idea behind the CPM, which is that both the emission and state sequences are conditioned by an additional input sequence \( Z_{1:T} \).

Listgarten et al. (2004) attained an alignment under their CPM by considering that the additional input sequence is an underlying ‘latent trace’ that represents a common representation of all the observed time series. Crucially, there is only a single latent trace in the model and all of the observed time series are aligned to the same trace. Each time series is modelled as an emission sequence of the CPM and the corresponding realisation of the state sequence is a mapping to the common latent trace. The latent trace has a much higher number of time points than the observed
time series (not shown in Definition 5.2), which allows the mapping of an observed time series to ‘slow down’ and ‘speed up’ relative to ‘latent time’ and hence align the observed time series to the latent trace.

Listgarten, Neal, Roweis, Puckrin & Cutler (2007) proposed an extension to CPMs where subgroups of the time series to be aligned may have systematic deviations from the common latent trace. Their extended model incorporates a common ‘parent’ trace in addition to a ‘child’ trace for each of the subgroups with systematic deviations. However, there is effectively still only a single latent trace in the model and all the time series that are modelled are still assumed to have the same basic qualitative shape.

Recall that we aim to obtain a common representation of each pair of expression profiles corresponding to each gene, not a single representation for all expression profiles of all genes. A single latent trace would be inappropriate for representing all of the expression profiles in the grapevine data. We would require a separate latent trace and hence a separate model for the pair of expression profiles corresponding to each gene. However, the CPM was developed for mass spectrometry and speech waveform data sets that contained around 10 replicate time series of approximately 800 time points, while we would correspondingly have 2 replicate profiles of approximately 18 time points. We conclude that the CPM would be an inappropriate model for our purposes.

### 5.3 Simple Alignment with an HMM

Lin, Kaminski & Bar-Joseph (2008) addressed the problem of alignment of gene expression profiles using HMMs that cannot be considered as an extension of a Pair HMM. They aimed at classifying patients into different classes of response to treatment based on the patient’s observed expression profiles. However, as the response to treatment may vary in time for different patients, an alignment of the observed expression profiles was taken into consideration in their model.

Lin et al. (2008) achieved their classification result by training distinct HMMs from labeled data and then using the multiple models for discrimination. They considered the alignment of the expression profiles as a completely separate problem and achieved this by constraining the transition matrix of the Markov chain component of the model. They considered HMMs such that the Markov chain component is constrained to be a ‘left-right’ model.

**Definition 5.3 Left-right HMM.** A *Left-right HMM* is a hidden Markov model \( \{X_{1:T}, S_{1:T}\} \) such that \( a_1 = 1 \), and \( a_{ij} \neq 0 \) if and only if \( j = i, i + 1 \) for \( i = 1, 2, \ldots, N \).
In a Left-right HMM, a state can never be revisited once it has been left and transitions away from a state may only occur to a single other state. Consider a Left-right HMM with \( N = 3 \) states and Markov parameters
\[
a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} a_{11} & 1-a_{11} & 0 \\ 0 & a_{22} & 1-a_{22} \\ 0 & 0 & 1 \end{bmatrix}.
\] (5.7)

The state transition diagram of this model is given in Figure 5.2. In a state transition diagram of a Markov chain, each vertex corresponds to a value that the state random variables can take while directed edges indicate the possible transitions between states with non-zero probability.

![State transition diagram](image)

Figure 5.2: The state transition diagram of a Left-right HMM with \( N = 3 \) states and Markov parameters (5.7).

By constructing the state transition diagram as in Figure 5.2, we can see that the transitions can only occur from left to right. Any realised state sequence must start in state 1, move to state 2, then move to state 3. Hence, the possible differences between realisations of the state sequence are the times at which the transitions occur.

Lin et al. (2008) interpreted the hidden states as ‘disease phases’ and so the left-right Markov structure allows different patients to progress through the same order of disease phases but at different times. Hence an alignment is achieved between the expression profiles of different patients by modelling the profiles as emission sequences of a Left-right HMM and consider at what time the transitions occur in the most likely corresponding realisation of the state sequence, the Viterbi path (Section 4.2).

The definition of a Left-right HMM can be altered to allow for less restrictive transitions between states while keeping the same alignment idea, for example allowing the ‘leapfrogging’ of states by setting \( a_{ij} \neq 0 \) if and only if \( j = i, i+1, i+2 \) for \( i = 1, 2, \ldots, N \). Schliep, Schönhuth & Steinhoff (2003) also considered alignment in such a way, however their main focus was a model based clustering method for expression profiles where each cluster is represented by an HMM. Schliep,

In a Left-right HMM, all realised state sequences must follow the same ‘left-right’ order. When modelling the grapevine data we want to interpret the hidden state sequences as representing distinct quantitative levels of the corresponding expression profiles. That is, capture the basic patterns of the pairs of expression profiles that may not necessarily share the same shape as any other pair. In order to achieve their alignment, Lin et al. (2008) constrained the transitions of the state sequence to the extent that all realised state sequences must share the same basic shape. Such a model would be inappropriate for the grapevine data.
Part II
Chapter 6

$L(t)$-fold HMMs

We define and describe an extension of an HMM that is obtained by increasing the number of emission random variables in the model. In practice, we still consider that the emission random variables are observed while the realisations of the state sequence are ‘hidden’. Hence we still have a ‘hidden Markov model’ and we additionally retain the notion that the Markov chain state sequence is the driving process.

Consider an HMM such that rather than a single emission random variable at each time point, there are $L(t)$ emission random variables. The points of difference from an HMM are that there are multiple emission random variables at each time point $t$ and in particular, a potentially different number $L(t)$ emission random variables, hence $L(t)$-fold HMM. Although there is an extensive literature on HMMs and $L(t)$-fold HMMs have been anticipated as a particular case of more general extensions (Yu 2010), our formulation is to our knowledge new. We present the extension in the generality required for our alignment application, which itself is the particular novelty.

We define and describe $L(t)$-fold HMMs, paying particular attention to the similarities they have with HMMs (Section 6.1), and give a parameterisation of the model (Section 6.2).

### 6.1 Definition

Let $L(t) \in \{1, 2, \ldots\}$ for $t = 1, 2, \ldots, T$. Write the collection of $L(t)$ random variables with common subscript $t$ as

$$X_t^{(1:L(t))} = \{X_t^{(1)}, X_t^{(2)}, \ldots, X_t^{(L(t))}\}$$
for $t = 1, 2, \ldots, T$. We write the total collection of $\sum_{t=1}^{T} L(t)$ random variables

$$X_{1:T} = \{X_{1}^{(1:L(1))}, X_{2}^{(1:L(2))}, \ldots, X_{T}^{(1:L(T))}\}.$$

We also write

$$X_{t}^{(l-1)} = \{X_{t}^{(1)}, \ldots, X_{t}^{(l-1)}, X_{t}^{(l+1)}, \ldots, X_{t}^{(L(t))}\}$$

for $t = 1, 2, \ldots, T$ and $l = 1, 2, \ldots, L(t)$, and

$$X_{t} = \{X_{t}^{(1:L(1))}, \ldots, X_{t-1}^{(1:L(t-1))}, X_{t+1}^{(1:L(t+1))}, \ldots, X_{T}^{(1:L(T))}\}$$

for $t = 1, 2, \ldots, T$. We use the same notation for the realisations of these random variables.

**Definition 6.1 $L(t)$-fold HMM.** An $L(t)$-fold HMM is a collection of random variables $\{X_{1:T}, S_{1:T}\}$ such that

$$\{X_{1:t-1}, S_{1:t-1}\} \perp \perp \{X_{t+1:T}, S_{t+1:T}\} \mid S_{t} \quad (6.1)$$

for $t = 2, 3, \ldots, T-1,$

$$X_{t}^{(1:L(t))} \perp \perp \{X_{\sim t}, S_{\sim t}\} \mid S_{t} \quad (6.2)$$

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

$$X_{t}^{(l)} \perp \perp X_{t}^{(l-1)} \mid S_{t} \quad (6.3)$$

for $t = 1, 2, \ldots, T$ and $l = 1, 2, \ldots, L(t)$.

We call $X_{1:T}$ the emission random variables and $S_{1:T}$ the state sequence of an $L(t)$-fold HMM. Conditional independence properties (6.1) and (6.2) are analogous to properties (3.1) and (3.2) of an HMM. The additional property (6.3) concerns only the $L(t)$ emission random variables at time $t$ and hence has no analogue under an HMM.

For the same reason as HMMs (Section 3.1), $L(t)$-fold HMMs also satisfy the separation condition (2.2) and hence Theorem 2.3 holds. Hence the collection of random variables $\{X_{1:T}, S_{1:T}\}$ is an $L(t)$-fold HMM if and only if the conditional independence graph of the collection is given in Figure 6.1. In Figure 6.1, we can clearly see that the state sequence is a Markov chain. We additionally verify that the underlying Markov chain state sequences is the driving process of the model as the vertex corresponding to each individual emission random variable only has a single
edge to the vertex corresponding to the state random variable at that time. That is, as for HMMs, each emission random variable is conditionally independent of all of other random variables given the state random variable at the corresponding time,

\[ X_t^{(l)} \perp \perp \{ X_{<t}, X_t^{(\sim l)}, S_t \} \mid S_t \]

for \( t = 1, 2, \ldots, T \) and \( l = 1, 2, \ldots, L(t) \). So we understand the conditional independence properties (6.1), (6.2) and (6.3) as a general Markov property and two stronger conditional independence properties of the emission random variables respectively.

Because of the correspondence between the properties (6.1) and (6.2) of an \( L(t) \)-fold HMM, and (3.1) and (3.2) of an HMM, there are a number of analogous conditional independence properties that hold for analogous collections of random variable under both models. That is, since any property that holds under an HMM follows by decomposition from (3.1) and (3.2), there exists an analogous property for an \( L(t) \)-fold HMM that follows from (6.1) and (6.2). For example, \( X_{1:t-1} \perp X_{t+1:T} \mid S_t \) for \( t = 2, 3, \ldots, T - 1 \) from (3.1). Analogously, \( X_{1:t-1} \perp X_{t+1:T} \mid S_t \) for \( t = 2, 3, \ldots, T - 1 \) from (6.1). We will use the fact that such analogous conditional independence properties hold in order to adapt the solutions to the associated HMM problems (Chapter 4) to obtain solutions to the analogous associated problems for \( L(t) \)-fold HMMs (Chapter 7).

It is clear from both Definition 6.1 and Figure 6.1 that an \( L(t) \)-fold HMM is equivalent to an HMM if \( L(t) = 1 \) for \( t = 1, 2, \ldots, T \). In that case, properties (6.1) and (6.2) are exactly those of Definition 3.1 and property (6.3) is meaningless. Given this, the fact that the state sequence is still a Markov chain and that the individual emission random variables have the same conditional independence structure under both models, it is not surprising that an \( L(t) \)-fold HMM is parameterised by the same set of parameters as an HMM.

Figure 6.1: Conditional independence graph of an \( L(t) \)-fold HMM.
6.2 Parameterisation

We consider the joint density of the emission random variables and the state sequence and show that an \( L(t) \)-fold HMM can be parameterised by the same set of HMM parameters \( \lambda \). The joint density of the emission random variables and state sequences of an \( L(t) \)-fold HMM is

\[
p(x_{1:T}, s_{1:T}) = p(s_1) \prod_{t=2}^{T} p(s_t | s_{t-1}) \left( \prod_{t=1}^{T} \prod_{l=1}^{L(t)} p(x_{t}^{(l)} | s_t) \right).
\]

This can be seen from the HMM density (Section 3.2) and as the analogous conditional independence properties hold.

As for an HMM, we can see that the density of the \( L(t) \)-fold HMM is the product of the Markov chain component of the model and the conditional emission densities. Recall that we consider the Markov chain state sequence to be discrete and time homogeneous. We consider that

\[
X_{t}^{(l)} | \{ S_t = j \} \sim N(\mu_j, \sigma_j^2)
\]

for \( t = 1, 2, \ldots, T \) and \( l = 1, 2, \ldots, L(t) \). Hence we require the same quantities that parameterise an HMM and so we take the \( L(t) \)-fold HMM parameters to be the same set \( \lambda \) as in Definition 3.2. We have the joint density

\[
p(x_{1:T}, s_{1:T}|\lambda) = a_{s_1} \prod_{t=2}^{T} a_{s_{t-1} s_t} \left( \prod_{t=1}^{T} \prod_{l=1}^{L(t)} b(x_{t}^{(l)} | \mu_{s_t}, \sigma_{s_t}^2) \right).
\]
Chapter 7

Associated Problems for $L(t)$-fold HMMs

Analogous to the HMM case, there are three associated problems for $L(t)$-fold HMMs and the solutions to these problems allow for the model to be used in practice:

1. **Evaluation of the Marginal Emission Density** (Section 7.1)

   Given observed emission random variables of an $L(t)$-fold HMM with known parameters $\lambda$, evaluate the marginal emission density at these observations.

2. **Most Likely Corresponding Realised State Sequence** (Section 7.2)

   Given observed emission random variables of an $L(t)$-fold HMM with known parameters $\lambda$, find the most likely corresponding realisation of the state sequence.

3. **Maximum Likelihood Estimator of the Parameters** (Section 7.3)

   Given observed emission random variables of $K$ independent and identically distributed $L(t)$-fold HMMs, find the maximum likelihood estimator of the parameters $\lambda$.

   In the following, we show that the standard solutions to the associated HMM problems can be adapted in order to obtain solutions for the $L(t)$-fold HMM problems.
7.1 Evaluation of the Marginal Emission Density

Suppose we have observed the emission random variables of an $L(t)$-fold HMM with known parameters $\lambda$. We evaluate the marginal emission density at these observations. As in Section 4.1, evaluating the marginal emission density

$$p(x_1:T|\lambda) = \sum_{s_1=1}^{N} \cdots \sum_{s_t=1}^{N} a_{s_1} \prod_{t=2}^{T} a_{s_t-1,s_t} \left( \prod_{l=1}^{T} L(l) \prod_{t=1}^{T} b(x_t|\mu_{s_t},\sigma_{s_t}^2) \right)$$

is computationally infeasible as such a procedure requires in the order of $N^T$ calculations. This problem is solved using analogous forward and backward functions from Section 4.1, the only difference is that there is an additional product over $l = 1,2,\ldots,L(t)$ for each $t = 1,2,\ldots,T$ in any density function accounting for the $L(t)$ emission random variables at time $t$, which simply carries through.

**Definition 7.1 $L(t)$-fold Forward and Backward Functions.** Define the $L(t)$-fold forward function

$$\alpha_t(j|\lambda) = p(x_{1:t},S_t = j|\lambda)$$

and the $L(t)$-fold backward function

$$\beta_t(j|\lambda) = p(x_{t+1:T}|S_t = j,\lambda).$$

**Theorem 7.1 $L(t)$-fold Forward-Backward Evaluation (Solution to the First Problem).**

Let the collection of random variables $\{X_{1:T},S_{1:T}\}$ be an $L(t)$-fold HMM with known parameters $\lambda$. Given observed emission random variables, then

$$p(x_1:T|\lambda) = \sum_{j=1}^{N} \alpha_t(j|\lambda) \beta_t(j|\lambda)$$

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

We show that evaluating the marginal density using Theorem 7.1 is a considerable decrease in computational expense compared to marginalising the joint density. We consider analogous results as in Section 4.1 for the evaluation of the $L(t)$-fold forward and backward functions.
Lemma 7.2 $L(t)$-fold Forward Evaluation. For the $L(t)$-fold forward function (7.1),

$$\alpha_1(i|\lambda) = \prod_{l=1}^{L(1)} b(x_1^{(l)}|\mu_i, \sigma_i^2) a_i,$$

and

$$\alpha_t(j|\lambda) = \prod_{l=1}^{L(t)} b(x_t^{(l)}|\mu_j, \sigma_j^2) \sum_{i=1}^{N} a_{ij} \alpha_{t-1}(i|\lambda)$$

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

Proof of Lemma 7.2. This follows from the proof of Lemma 4.2 but where we use the definition of the forward function given by (7.1) as the analogous necessary conditional independence properties hold (Section 6.1).

We again consider trellis diagrams to visualise the evaluation of the $L(t)$-fold forward and backward functions. For the $L(t)$-fold forward function, we obtain the same trellis diagram as in Figure 4.1 except we use the definition of the forward function given by (7.1). We still move forward through the trellis, evaluating the vertices for each time point of the trellis in turn and by using the quantities corresponding to the vertices of the previous time point. It can be seen that the evaluation of (7.1) for all $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N$ requires $NL(1) + \sum_{t=2}^{T} N(N + L(t))$ multiplications and $N(N - 1)(T - 1)$ additions in total.

Lemma 7.3 $L(t)$-fold Backward Evaluation. For the $L(t)$-fold backward function (7.2),

$$\beta_T(j|\lambda) = 1$$

for $j = 1, 2, \ldots, N$ and

$$\beta_{t-1}(i|\lambda) = \sum_{j=1}^{N} \beta_t(j|\lambda) \prod_{l=1}^{L(t)} b(x_t^{(l)}|\mu_j, \sigma_j^2) a_{ij}$$

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

Proof of Lemma 7.3. This follows from the proof of Lemma 4.3 but where we use the definition of the backward function given by (7.2) as the analogous necessary conditional independence properties hold.
CHAPTER 7. ASSOCIATED PROBLEMS FOR $L(t)$-FOLD HMMS

It can be seen that to completely evaluate (7.2) for all $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N$ requires \( \sum_{t=2}^{T}(L(t) + 1)N^2 \) multiplications and \( N(N - 1)(T - 1) \) additions.

Proof of Theorem 7.1. This follows by the same arguments used in the proof of Theorem 4.1 as the analogous necessary conditional independence properties hold.

Hence, to evaluate the marginal emission density using Theorem 7.1 requires the evaluation of (7.1) up to time $t$ and (7.2) back to time $t$. This requires $NL(1) + \sum_{t'=2}^{t}N(N + L(t')) + \sum_{t'=1}^{T}(L(t') + 1)N^2$ multiplications and $N(N - 1)(t - 1) + N(N - 1)(T - t)$ additions. We then require an additional $N$ multiplications and $N - 1$ additions to evaluate (7.3). Once again, this is a considerable reduction in computational expense compared to marginalising the joint density directly.

7.2 Most Likely Corresponding Realised State Sequence

Suppose we have observed emission random variables of an $L(t)$-fold HMM with known parameters $\lambda$. We find the most likely corresponding realisation of the state sequence. The overall most likely such sequence is

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}} p(s_{1:T}|x_{1:T}).$$

By the same arguments as in Section 4.2, we equivalently have

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}} p(x_{1:T}, s_{1:T})$$

and the problem that we cannot feasibly evaluate the joint density for each possible realisation of the state sequence. We work through the analogous solution using the $L(t)$-fold Viterbi function where once again there is just an additional product over $l = 1, 2, \ldots, L(t)$ for for each $t = 1, 2, \ldots, T$ in any density function accounting for the $L(t)$ emission random variables at time $t$, which simply carries through.

Definition 7.2 $L(t)$-fold Viterbi Function. Define the $L(t)$-fold Viterbi function

$$v_t(j) = \max_{s_{1:t-1}} p(x_{1:t}, s_{1:t-1}, S_t = j).$$  (7.4)
Theorem 7.4 $L(t)$-fold Viterbi Path (Solution to the Second Problem). Let the collection of random variables \(\{X_{1:T}, S_{1:T}\}\) be an $L(t)$-fold HMM with known parameters $\lambda$. Given observed emission random variables, the overall most likely corresponding realisation of the state sequence is $\hat{s}_{1:T}$ where

$$
\hat{s}_T = \arg\max_j [v_T(j)]
$$

and

$$
\hat{s}_{t-1} = \arg\max_i [v_{t-1}(i)a_{is_t}]
$$

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

Once again we call $\hat{s}_{1:T}$ the Viterbi path corresponding to the observed emission random variables.

Lemma 7.5 $L(t)$-fold Viterbi Evaluation. For the $L(t)$-fold Viterbi function (7.4),

$$
v_1(i) = \prod_{l=1}^{L(1)} b(x_1^{(l)}|\mu_i, \sigma_i^2)a_i
$$

and

$$
v_t(j) = \max_i \left[v_{t-1}(i)a_{ij}\right] \prod_{l=1}^{L(t)} b(x_t^{(l)}|\mu_j, \sigma_j^2)
$$

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

Proof of Lemma 7.5. This follows from the proof of Lemma 4.5 where we use the definition of the Viterbi function given by (7.4) as the analogous necessary conditional independence properties hold.

Proof of Theorem 7.4. This follows by the same arguments as used in the proof of Theorem 4.4, the only difference is that we consider the definition of the Viterbi function given by (7.4).
Hence, given (7.4) evaluated for all \( t = 1, 2, \ldots, T \) and \( j = 1, 2, \ldots, N \), the calculation of the \( L(t) \)-fold Viterbi path requires \( N(T - 1) \) multiplications and \( T \) \( \text{argmax} \) operations over \( N \) quantities. This is once again a considerable decrease in computational expense.

### 7.3 Maximum Likelihood Estimator of the Parameters

Suppose we have observed emission random variables of \( K \) independent and identically distributed \( L(t) \)-fold HMMs. We find the maximum likelihood estimator (MLE) of the model parameters \( \lambda \).

We write the emission random variables from the \( k^{th} \) model at time \( t \) as

\[
X_t^{(k)} = X_t^{(k)(1)}, X_t^{(k)(2)}, \ldots, X_t^{(k)(L(t))}
\]

and the total collection of emission random variables from the \( k^{th} \) model

\[
X_{1:T}^{(k)} = \{X_1^{(k)(1):L(1)}, X_2^{(k)(1):L(2)}, \ldots, X_T^{(k)(1):L(T)}\}.
\]

Then we write the total collection of emission random variables from all \( K \) models

\[
X_{1:T}^{(1:K)} = \{X_1^{(1)}, X_2^{(2)}, \ldots, X_K^{(K)}\}
\]

and the total collection of emission random variables except those from the \( k^{th} \) model

\[
X_{1:T}^{(\sim k)} = \{X_1^{(1)}, \ldots, X_{k-1}^{(k-1)}, X_{k+1}^{(k+1)}, \ldots, X_T^{(K)}\}
\]

for \( k = 1, 2, \ldots, K \). We use the same notation for the collection of \( K \) state sequences and for collections of realisations of these random variables. Hence the independence assumption is

\[
\{X_{1:T}^{(k)}, S_{1:T}^{(k)}\} \perp \perp \{X_{1:T}^{(\sim k)}, S_{1:T}^{(\sim k)}\}
\]

where \( \{X_{1:T}^{(k)}, S_{1:T}^{(k)}\} \) is an \( L(t) \)-fold HMM with unknown parameters \( \lambda \) for \( k = 1, 2, \ldots, K \).

For the same reason as in Section 4.3, we cannot obtain analytical MLE solutions from the marginal emission random variable log-likelihood. However, the solution is given by the analogous Baum-Welch algorithm.
Algorithm 7.1 $L(t)$-fold Baum-Welch Algorithm (Solution to the Third Problem). Let the collection of random variables $\{X_{1:T}^{(k)}, S_{1:T}^{(k)}\}$ be an independent and identically distributed $L(t)$-fold HMMs for $k = 1, 2, \ldots, K$. Given observed emission random variables, to obtain the maximum likelihood estimate of the parameters $\lambda$:

1. Obtain initial estimate of the parameters, $\lambda^{[0]}$.

2. At the $n$th iteration, calculate

   $$a_i^{[n]} = \frac{1}{K} \sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]}),$$

   (7.5)

   $$a_{ij}^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=2}^{T} \zeta_t^{(k)}(i,j|\lambda^{[n-1]})}{\sum_{k=1}^{K} \sum_{t=2}^{T} \zeta_{t-1}^{(k)}(i|\lambda^{[n-1]})},$$

   (7.6)

   $$\mu_j^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{l=1}^{L(l)} \zeta_t^{(k)}(j|\lambda^{[n-1]})(\mathbb{E}_t^{(l)})}{\sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{l=1}^{L(l)} \zeta_t^{(k)}(j|\lambda^{[n-1]})},$$

   (7.7)

   and

   $$(\sigma_j^2)^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{l=1}^{L(l)} \zeta_t^{(k)}(j|\lambda^{[n-1]})(x_t^{(l)})^2}{\sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{l=1}^{L(l)} \zeta_t^{(k)}(j|\lambda^{[n-1]})},$$

   (7.8)

   for $i, j = 1, 2, \ldots, N$ where

   $$\zeta_t^{(k)}(j|\lambda^{[n-1]}) = p(S_t^{(k)} = j|X_{1:T}^{(k)}, \lambda^{[n-1]})$$

   (7.9)

   and

   $$\zeta_t^{(k)}(i, j|\lambda^{[n-1]}) = p(S_t^{(k)} = i, S_{t-1}^{(k)} = j|X_{1:T}^{(k)}, \lambda^{[n-1]}).$$

   (7.10)

3. Return to Step 2 for a specified number of iterations or until some convergence criterion is satisfied.

As before, the $L(t)$-fold Baum-Welch algorithm is a special case of the EM algorithm where we take the realisations of the emission random variables to be the observed data and the corresponding realisations of the state sequence to be the unobserved data. The only difference from Section
4.3 is that there is an additional sum over \(l = 1, 2, \ldots, L(t)\) for each \(t = 1, 2, \ldots, T\) in the full data log-likelihood accounting for the \(L(t)\) emission random variables at each time. This sum is simply carried through when finding the analytical parameter updates and appears in the emission distribution parameter updates (7.7) and (7.8).

We have the joint density

\[
p(x_{1:T}^{(1:K)}, s_{1:T}^{(1:K)}|\lambda) = \prod_{k=1}^{K} \left[ p(x_{1:T}^{(k)}, s_{1:T}^{(k)}|\lambda) \right]
\]

and the full data log-likelihood

\[
l(\lambda|x_{1:T}^{(1:K)}, s_{1:T}^{(1:K)}) = \log \left[ \prod_{k=1}^{K} \prod_{t=2}^{T} a_{s_{t-1:s_t}^{(k)}} \left( \prod_{l=1}^{L(t)} b(x_{t}^{(k)(l)}|\mu_{s_t^{(k)}}, \sigma_{s_t^{(k)}}^2) \right) \right]
\]

\[
= \sum_{k=1}^{K} \log[a_{s_1^{(k)}}] + \sum_{k=1}^{K} \sum_{t=2}^{T} \log[a_{s_{t-1:s_t}^{(k)}}]
\]

\[
+ \sum_{k=1}^{K} \sum_{l=1}^{L(t)} \sum_{t=1}^{T} \log[b(x_{t}^{(k)(l)}|\mu_{s_t^{(k)}}, \sigma_{s_t^{(k)}}^2)].
\]

The EM algorithm (Algorithm B.1) requires that at the \(n\)th iteration we find

\[
\lambda^{[n]} = \operatorname{argmax}_\lambda Q(\lambda|\lambda^{[n-1]})
\]

where the objective function is

\[
Q(\lambda|\lambda^{[n-1]}) = E_{S_{1:T}^{(1:K)}} \left[ l(\lambda|x_{1:T}^{(1:K)}, S_{1:T}^{(1:K)}) \right| x_{1:T}^{(1:K)}, \lambda^{[n-1]}].
\] (7.11)

**Theorem 7.6** \(L(t)\)-fold Analytic Parameter Updates. Given current estimate \(\lambda^{[n-1]}\), the objective function (7.11) is locally maximised at \(\lambda^{[n]}\) given by (7.5), (7.6), (7.7) and (7.8).

**Proof of Theorem 7.6.** This follows from the proof of Theorem 4.6 where we use the definition of the weights \(\zeta_i^{(k)}(i|\lambda^{[n-1]})\) and \(\xi_t^{(k)}(i, j|\lambda^{[n-1]})\) given by (7.9) and (7.10) respectively. The extra summation over \(l = 1, 2, \ldots, L(t)\) for each \(t = 1, 2, \ldots, T\) that is not present in the result of Theorem 4.6 simply carries through when rewriting the objective function. When maximising the objective function, the parameters \(B\) that we maximise over are independent of this summing index and hence all of the arguments from the proof of Theorem 4.6 still follow through. □
We additionally have an analogous result for the evaluation of the weights (7.9) and (7.10) so that the $L(t)$-fold Baum-Welch algorithm can be used in practice.

**Lemma 7.7 Evaluation of $L(t)$-fold Weights.** For the $L(t)$-fold weights $\zeta_t^{(k)}(j|\lambda)$ and $\xi_t^{(k)}(i,j|\lambda)$ given by (7.9) and (7.10) respectively,

$$
\zeta_t^{(k)}(j|\lambda) = \frac{\alpha_t^{(k)}(j|\lambda)\beta_t^{(k)}(j|\lambda)}{\sum_{j=1}^{N} \alpha_t^{(k)}(j|\lambda)\beta_t^{(k)}(j|\lambda)}
$$

and

$$
\xi_t^{(k)}(i,j|\lambda) = \frac{\alpha_{t-1}^{(k)}(i|\lambda)\prod_{l=1}^{L(t)} b(x_t^{(k)}(l)|\mu_j, \sigma_j^2)\beta_t^{(k)}(j|\lambda)\alpha_{ij}}{\sum_{j=1}^{N} \alpha_t^{(k)}(j|\lambda)\beta_t^{(k)}(j|\lambda)}
$$

where

$$
\alpha_t^{(k)}(j|\lambda) = p(x_t^{(k)}, S_t^{(k)} = j|\lambda)
$$

and

$$
\beta_t^{(k)}(j|\lambda) = p(x_{t+1:T}, S_t^{(k)} = j, \lambda).
$$

**Proof of Lemma 7.7.** This follows from the proof of Lemma 4.7 but where we use the definitions of the $L(t)$-fold weights (7.9) and (7.10), and forward and backward functions (7.1) and (7.2), and as the analogous necessary conditional independence properties hold.
Chapter 8

Alignment with $L(t)$-fold HMMs

We demonstrate how an alignment between different length sequences can be achieved under an $L(t)$-fold HMM. We first consider simulated data from an example model to illustrate such an alignment (Section 8.1). Then we present the proposed model for the grapevine data and explain how the alignment between the Willunga and Clare vineyards is achieved (Section 8.2).

8.1 An Example Model

Consider the conditional independence graph given in Figure 8.1. Recall that a conditional independence graph can be used to define an $L(t)$-fold HMM (Section 6.1). The graph in Figure 8.1 defines the $L(t)$-fold HMM with

$$L(t) = \begin{cases} 
1 & \text{if } t = 5 \text{ or } 13 \\
2 & \text{otherwise}
\end{cases} \quad (8.1)$$

for $t = 1, 2, \ldots, 19$. This example $L(t)$-fold HMM can be considered to have two emission sequences, one plotted above and the other below the state sequence in Figure 8.1. Note that the second emission sequence has two gaps in it as there are no emission random variables $X_5^{(2)}$ and $X_{13}^{(2)}$ in the model.
Consider an $N = 3$ state $L(t)$-fold HMM with conditional independence graph given in Figure 8.1, Markov parameters

$$a = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}, \quad A = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \quad (8.2)$$

and emission distributions

$$X_t^{(l)} \mid \{S_t = j\} \sim \begin{cases} N(1, 0.2) & \text{if } j = 1 \\ N(2, 0.2) & \text{if } j = 2 \\ N(3, 0.2) & \text{if } j = 3. \end{cases} \quad (8.3)$$

Simulated data from this model is plotted in Figure 8.2. In addition to the simulated emission sequences, the corresponding simulated state sequence is also shown. Both simulated emission sequences are aligned to the simulated state sequence. Although we always consider that the state random variables take values in the first $N$ natural numbers, in general these values do not have any relationship to the corresponding emission distributions. The alignment in Figure 8.2 is especially apparent as we have set $\mu_j = j$ for $j = 1, 2, 3$ and hence the observed values of all sequences are quantitatively similar at each time point.

In Figure 8.3 we plot the concatenated length 17 simulated emission sequence with no gaps in addition to the length 19 simulated emission sequence from Figure 8.2. The two sequences in Figure 8.3 still share the same basic shape but now have different lengths and the timing of the features are different. That is, the simulated emission sequences as plotted in Figure 8.3 are aligned in time as plotted in Figure 8.2.
CHAPTER 8. ALIGNMENT WITH L(t)-FOLD HMMS

Figure 8.2: Simulated emission random variables and state sequence of the \( L(t) \)-fold HMM with conditional independence graph given in Figure 8.1 and parameters given by (8.2) and (8.3). The simulated emission sequence \( x_{1:19}^{(1)} \) is plotted as blue stars (\( \times \)), the simulated emission random variables \( x_{1:4}^{(2)}, x_{6:12}^{(2)} \) and \( x_{14:19}^{(2)} \) are plotted as red stars (\( \ast \)), and the simulated state sequence \( s_{1:19} \) is plotted as black circles (\( \circ \)).

Figure 8.3: Simulated emission sequences as in Figure 8.2 but where the simulated emission random variables \( x_{1:4}^{(2)}, x_{6:12}^{(2)} \) and \( x_{14:19}^{(2)} \) have been concatenated into a single realised emission sequence of length 17 with no gaps.

We call the values of \( t \) for which \( L(t) = 1 \) the gap positions. In this example model, the gap positions are 5 and 13. The gap positions produce the timing differences between the simulated emission sequences seen in Figure 8.3 and different values of the gap positions would have resulted in different timing differences. The problem we will ultimately consider is the estimation of the gap positions from data such as that plotted in Figure 8.3 in order to obtain the alignment of the sequences as shown in Figure 8.2.
8.2 Model for the Grapevine Data

Denote $1 < G_1 < G_2 \leq T = 19$ as the two gap positions. Consider an $L(t)$-fold HMM where

$$L(t) = \begin{cases} 1 & \text{if } t = G_1 \text{ or } G_2 \\ 2 & \text{otherwise} \end{cases}$$

(8.4)

for $t = 1, 2, \ldots, 19$. The example model presented in Section 8.1 is an $L(t)$-fold HMM satisfying (8.4) where $G_1 = 5$ and $G_2 = 13$. The gap positions $G_1$ and $G_2$ can be considered as additional parameters of the model and are estimated from data.

Consider an $L(t)$-fold HMM satisfying (8.4) where the length 19 emission sequence is labelled $W_{1:19}$ and the length 17 emission sequence is labelled $C_{1:17}$. We will model the pairs of profiles from the Willunga and Clare vineyards as the sequences of random variables $W_{1:19}$ and $C_{1:17}$ respectively. The different length emission sequences $W_{1:19}$ and $C_{1:17}$ are aligned under the model in that aligned time points of the expression profiles are modelled by emission random variables that are are conditioned by the same state random variable.

Recall that under a Pair HMM, insert elements are added to observed genomic sequences in order to align two such sequences (Section 5.1). The gap positions of the alignment $L(t)$-fold HMM can be considered as analogous to the insert elements of a Pair HMM. However, the gap positions are not determined by the underlying Markov chain structure of the model like the inserts are for a Pair HMM, but are rather fixed components of the alignment model. The ‘additional information’ that aligns the Willunga and Clare profiles is the gap positions and this allows the Markov chain component of the model to represent distinct quantitative levels of expression.

More generally, the $k^{th}$ pair of expression profiles are modelled as the sequences of emission random variables $W_{1:19}^{(k)}$ and $C_{1:17}^{(k)}$ of independent and identically distributed $L(t)$-fold HMMs satisfying (8.4) for $k = 1, 2, \ldots, K$. The log-likelihood function is

$$I(\lambda|w_{1:19}^{(k)}, c_{1:17}^{(k)}) = \log \left[ \frac{p(w_{1:19}^{(k)}, c_{1:17}^{(k)}|\lambda)}{p(w_{1:19}^{(k)}, c_{1:17}^{(k)}|\lambda)} \right]$$

$$= \log \left[ \sum_{s_1^{(k)}=1}^N \sum_{s_2^{(k)}=1}^N \cdots \sum_{s_{19}^{(k)}=1}^N p(w_{1:19}^{(k)}, c_{1:17}^{(k)}, s_{1:19}^{(k)}|\lambda) \right]$$

$$= \log \left[ \sum_{s_1=1}^N \sum_{s_2=1}^N \cdots \sum_{s_{19}=1}^N a_{s_1} \left( \prod_{t=2}^{19} b(w_t^{(k)}|\mu_{s_t}, \sigma^2_{s_t}) \right) \left( \prod_{t=1}^{17} b(c_t^{(k)}|\mu_{s_t'}, \sigma^2_{s_t'}) \right) \right]$$

(8.5)
where

\[ t' = \begin{cases} 
  t & \text{for } t = 1, 2, \ldots, G_1 - 1 \\
  t + 1 & \text{for } t = G_1, G_1 + 1, \ldots, G_2 - 2 \\
  t + 2 & \text{for } t = G_2 - 1, G_2, \ldots, 17.
\end{cases} \]

The gap positions \( G_1 \) and \( G_2 \) are common for all pairs of profiles and so represent an overall alignment between the Willunga and Clare vineyards. An individual alignment for each pair would generally not be unique and the estimated gap positions would be more heavily influenced by random variability within that pair. By aggregating the information for all pairs, the estimated gap positions are more likely to represent a meaningful alignment between the two vineyards. The gap positions are constrained \( 1 < G_1 < G_2 \leq 19 \) so that the first time point of the profiles align, although not necessarily the last. This is simply due to the fact that when running the time course experiment, determining the end of the development cycle of the grape berries was more ambiguous than determining the start (Appendix A).

Under the proposed model, each pair of aligned profiles corresponding to the same gene will be modelled as the emission random variables corresponding to a single state sequence. For each pair of aligned profiles, the corresponding most likely realised state sequence, the \( L(t) \)-fold Viterbi path, can be obtained under the model (Section 7.2). The Viterbi paths can be interpreted as a possible common representation of the gene expression from the Willunga and Clare vineyards over the development of the grape berries.

Note that there is a somewhat unsatisfactory aspect of the model in that by considering multiple emission sequences under an \( L(t) \)-fold HMM, any of the emission random variables at the same time point can be swapped between the defined emission sequences and yet the model will retain the same distribution. To address this would require the introduction of correlation between consecutive emission random variables in the same sequence, which is beyond the scope of this project.

Additionally, note that the conditional emission densities under an \( L(t) \)-fold HMM are assumed to be Gaussian (Section 6.2) and so have the entire real number line as their domain. Due to the scaling of the grapevine data (Appendix A), the range of observed expression levels is \([0,1]\) with a large proportion of expression levels equal to 0 or 1 (Figure 8.4). So by our formulation of the model, a large proportion of the estimated conditional emission densities may lie beyond the range of the grapevine data. Note however that this feature of the model does not stop us from fitting the model nor will it cause any contradictions when we do so.
The number of states $N$ of an $L(t)$-fold HMM needs to be fixed prior to the estimation of the model parameters. The choice of $N$ is a problem in general for fitting HMMs and the standard approach is that the number of states should be chosen so as to give physical meaning to the states (Rabiner 1989). We wish to interpret the states as representing different levels of expression and want at least ‘low’, ‘medium’ and ‘high’ levels. Additionally, given the way in which the expression profiles are scaled, we will add two ‘boundary’ states to account for the high density of expression observations at either end of the range of observed expression levels (Figure 8.4). This will give us a total of $N = 5$ states for the alignment $L(t)$-fold HMM for the grapevine data.
Chapter 9

Model Fitting Methodology

We fit the alignment $L(t)$-fold HMM satisfying (8.4) by maximising the log-likelihood of the model with respect to the model parameters $\lambda$ and the gap positions $G_1$ and $G_2$.

If the gaps were given, the MLE of $\lambda$ can be obtained using the $L(t)$-fold Baum-Welch algorithm (Algorithm 7.1), hereafter just the Baum-Welch algorithm. As the gap positions are constrained $1 < G_1 < G_2 \leq 19$, there are $\binom{18}{2} = 153$ distinct values the gap positions can take. In principle we would train the alignment $L(t)$-fold HMM for each possible value of the gap positions and then take the estimated parameters and gap positions with overall maximum likelihood. However, we do not implement such a procedure due to its computational expense.

In the following we propose a less computationally expensive model fitting methodology for the alignment model (Section 9.1). We then illustrate the model fitting methodology in an example case by fitting the model to simulated data (Section 9.2).

9.1 Model Fitting Algorithm

In practice we will fit the $L(t)$-fold HMM satisfying (8.4) using a procedure incorporating both the Baum-Welch algorithm, to estimate of the model parameters $\lambda$, as well as a maximum likelihood estimation procedure for the gap positions. We propose the following algorithm to fit the alignment model and discuss each step in detail below.

Algorithm 9.1 Fitting the $L(t)$-fold HMM satisfying (8.4).

1. Obtain initial estimate of the parameters $\lambda^{[0]}$ and initial estimate of the gap positions, $G_1^{[0]}$ and $G_2^{[0]}$. 

64
2. Update the estimate of $\lambda$ using the Baum-Welch algorithm initialised by the current estimate of $\lambda$ and holding the current estimate of the gap positions fixed.

3. Update the estimate of the gap positions by finding the maximum likelihood gap positions given the current estimate of $\lambda$.

4. Return to Step 2 until the update estimate of the gap positions does not change.

Recall that for the EM algorithm, and hence the Baum-Welch algorithm, the initial estimate of the parameters affects the convergence of the log-likelihood values (Appendix B). Hence in general, the Baum-Welch algorithm is run a number of times in practice with randomly initialised estimates of the parameters for each run. Since our model fitting procedure is built around the Baum-Welch algorithm in Step 2, we will run Algorithm 9.1 multiple times in practice.

The Baum-Welch algorithm is entirely contained within Step 2 of Algorithm 9.1. In the first iteration, the Baum-Welch algorithm is initialised by the parameter estimates obtained in Step 1. In each subsequent iteration, the Baum-Welch algorithm is initialised by the current estimate of the parameters $\lambda$. The convergence criteria for stopping the Baum-Welch algorithm in Step 2 is a change in log-likelihood between Baum-Welch iterations that is less than $10^{-6}$.

In Step 3 the most likely gap positions are found by exhaustively evaluating the log-likelihood for all possible gap positions. Hence, no step of Algorithm 9.1 decreases the log-likelihood of the estimated parameters and gap positions under the model. The whole procedure can be thought of as a single implementation of the Baum-Welch algorithm that is systematically interrupted in order to also update the estimate of the gap positions.

### 9.2 Simulation Example

To illustrate the model fitting methodology and investigate whether it will work in practice, we use Algorithm 9.1 to fit an alignment $L(t)$-fold HMM to simulated data. Consider simulated data from the $L(t)$-fold HMM with conditional independence graph given in Figure 8.1 and parameters given by (8.2) and (8.3). We simulate $K = 100$ observations from this model and where the information about the gap positions has been removed. That is, the simulated data set is 100 pairs of observed emission sequences, one each of length 19 and 17, such as those given in Figure 8.3.

As $\lambda$ is known in this example, before considering Algorithm 9.1, we can first estimate the gap positions using maximum likelihood. That is, Step 3 of Algorithm 9.1 but where we use the true parameter values $\lambda$ given by (8.2) and (8.3). Figure 9.1 is a heat map of the log-likelihood of the
true model parameters $\lambda$ given the simulated data for each possible value of the gap positions. We can see that the log-likelihood peaks at the true values of the gap positions, $G_1 = 5$ and $G_2 = 13$, and monotonically decreases as the values of the gap positions move further from the truth. This suggests that the gap positions are well defined by the log-likelihood surface in this case.

![Figure 9.1: Heat map of the log-likelihood of the true model parameters $\lambda$ given the simulated data evaluated for each possible value of the gap positions.](image)

We now consider simultaneously estimating both the gap positions and parameters $\lambda$ using Algorithm 9.1. The modelling procedure was run 20 times where all computation was performed in MATLAB (R2009b) using the HMM Toolbox given by Murphy (2005). The HMM Toolbox has the necessary code to fit HMMs, which we modified (Chapter 7) in order to be able to fit $L(t)$-fold HMMs.

For each run we used the initialisation $\lambda^{[0]}$ given by

$$
a_i^{[0]} = \frac{U_i}{\sum_{i=1}^{3} U_i} \quad \text{and} \quad a_{ij}^{[0]} = \frac{U_{ij}}{\sum_{j=1}^{3} U_{ij}}
$$

where $U_i \sim U(0, 1)$ for $i = 1, 2, 3$ and $U_{ij} \sim U(0, 1)$ for $i, j = 1, 2, 3$,

$$
\mu_j^{[0]} \sim U(0, 4) \quad \text{and} \quad (\sigma_j^2)^{[0]} = 0.1
$$
for $j = 1, 2, 3$. We used the initialisation $G_1^{[0]}$ and $G_2^{[0]}$ given by randomly choosing a pair of possible values $1 < G_1 < G_2 \leq 19$ such that

$$p(G_1^{[0]} = G_1, G_2^{[0]} = G_2) = \frac{1}{\binom{19}{2}}.$$ 

We assume no prior information about any of the parameter values and so the initialisation of the Markov parameters, means of the emission distributions and gap positions are uniformly distributed over some appropriate range. Recall that since $1 < G_1 < G_2 \leq 19$, there are $\binom{18}{2} = 153$ distinct values the gap positions can take. The initialisation of the Markov parameters are also scaled to obey the stochastic constraint on these parameter values. The initial estimate of the variances of the emission distributions are simply set to a common fixed value. We set the maximum number of Baum-Welch iterations in Step 2 to be 1000 and the maximum number of overall iterations to be 10. We count the iterations of the Baum-Welch algorithm so that there are up to a total of $1000 \times 10 = 10000$ iterations for a single run of Algorithm 9.1.

Figure 9.2: Log-likelihood by iteration for a single run of Algorithm 9.1 fitting an $L(t)$-fold HMM to simulated data.

To illustrate the typical behaviour of the estimation process, consider the graph of log-likelihood by iteration for a single run of Algorithm 9.1 given in Figure 9.2. The first 84 iterations correspond to the Baum-Welch algorithm using the randomly initialised estimate of the model parameters.
and randomly initialised gap positions $G_1^{[0]} = 3$ and $G_2^{[0]} = 8$. At iteration 85, the Baum-Welch algorithm satisfies the convergence criterion. At this point, the estimate gap positions are updated (Step 3 of Algorithm 9.1) to $G_1^{[85]} = 5$ and $G_2^{[85]} = 13$. Then the Baum-Welch algorithm is applied again, inputting the current estimated parameters and updated gap positions. At iteration 114, the Baum-Welch algorithm satisfies the convergence criterion and the estimate gap positions are again updated to $G_1^{[114]} = 5$ and $G_2^{[114]} = 13$. Since the updated estimate gap positions remain the same, the entire process halts as the outer convergence criterion (Step 4) is now satisfied.

In Figure 9.3, we plot the log-likelihood by iteration for the 20 runs of the modelling procedure in addition to histograms of other related quantities. The top left plot is the log-likelihood by iteration for all 20 runs. All runs have the same general log-likelihood by iteration shape as plotted in Figure 9.2, although the ‘jump’ and end of the process occur at different iterations. The same final estimated gap positions $\hat{G}_1 = 5$ and $\hat{G}_2 = 13$ were attained in all 20 runs and were attained in the first instance of updating the estimate of the gap positions. In all runs, the estimated gap positions did not change when updated a second time and so the process halted as the outer convergence criterion of Algorithm 9.1 is satisfied. In the bottom of Figure 9.3 we can see histograms of the initial and converged values of the log-likelihood for all 20 runs. The log-likelihood of the estimated parameters all converged to values within $10^{-3}$ of the maximum log-likelihood from a much broader spread of initial log-likelihoods.

The estimated parameters $\hat{\lambda} = \{\hat{a}, \hat{A}, \hat{B}\}$ that have maximum log-likelihood over all of the 20 runs are the Markov parameters

$$
\hat{a} = \begin{bmatrix}
0.2712 \\
0.4408 \\
0.2880
\end{bmatrix}, \quad \hat{A} = \begin{bmatrix}
0.6411 & 0.1688 & 0.1901 \\
0.1799 & 0.6446 & 0.1756 \\
0.1916 & 0.1999 & 0.6085
\end{bmatrix}
$$

and the estimated parameters of the emission distributions

$$
X_t^{(i)} \mid \{S_t = j\} \sim \begin{cases}
N(0.9989, 0.2058) & \text{if } j = 1 \\
N(1.9643, 0.2166) & \text{if } j = 2 \\
N(2.9694, 0.2082) & \text{if } j = 3.
\end{cases}
$$

The estimated parameters are very similar to the true parameter values given by (8.2) and (8.3). As the estimated gap positions were equal to the true values, we conclude that the model fitting methodology given in Algorithm 9.1 has worked very well for this simulated example.
Figure 9.3: Output of fitting the alignment $L(t)$-fold HMM to simulated data using 20 runs of Algorithm 9.1. Top: Log-likelihood by iteration for the 20 runs and a histogram of the total number of iterations for each run. Bottom: Histograms of the initial and converged log-likelihood values for the 20 runs.
Chapter 10

Alignment of the Grapevine Data

We fit the alignment $L(t)$-fold HMM to the grapevine data using the proposed modelling procedure given by Algorithm 9.1. We discuss the performance of the fitting procedure, present the maximum likelihood estimate of the model parameters and interpret the alignment (Section 10.1). Then we calculate common representations of the pairs of aligned profiles (Section 10.2).

10.1 Fitting the Model

We fit the alignment $L(t)$-fold HMM to the grapevine data using the procedure given in Algorithm 9.1. The entire model fitting procedure was run 100 times, taking the overall maximum likelihood parameter estimates over all runs. Once again, we have no prior information about any of the parameter values and so for each run we used the initialisation $\lambda^{[0]}$ given by

$$a_i^{[0]} = \frac{U_i}{\sum_{i=1}^{5} U_i} \quad \text{and} \quad a_{ij}^{[0]} = \frac{U_{ij}}{\sum_{j=1}^{5} U_{ij}}$$

where $U_i \sim U(0, 1)$ for $i = 1, 2, \ldots, 5$ and $U_{ij} \sim U(0, 1)$ for $i, j = 1, 2, \ldots, 5$,

$$\mu_j^{[0]} \sim U(0, 1) \quad \text{and} \quad (\sigma_j^2)^{[0]} = 0.01$$

for $j = 1, 2, \ldots, 5$. We use the same initial estimate of the gap positions $G_1^{[0]}$ and $G_2^{[0]}$ as given in Section 9.2. As the profiles in the grapevine data have been scaled so that all observed expression levels lie between 0 and 1 (Appendix A), and we have no other prior information, the initial estimates of the means of the emission distributions are set to have standard uniform distributions.
CHAPTER 10. ALIGNMENT OF THE GRAPEVINE DATA

Figure 10.1: Output of fitting the alignment $L(t)$-fold HMM to the grapevine data using 100 runs of Algorithm 9.1. Top: Log-likelihood by iteration for the 100 runs and a histogram of the total number of iterations for each run. Bottom: Histograms of the initial and converged log-likelihood values for the 100 runs.
The initial estimates of the variances are set to a common fixed value. Additionally, the estimates of the variances are updated so that at the \( n \)th iteration, \((\sigma^2_j)^{[n]} \geq 0.001\) for \( j = 1, 2, \ldots, 5 \). This is a necessary constraint so that the update estimates do not get too close to zero and cause computational difficulties. The reason this is possible is due to high density of observed expression levels equal to 0 and 1 (Figure 8.4). As before, we set the maximum number of Baum-Welch iterations in Step 2 to be 1000 and the maximum number of overall iterations to be 10.

Figure 10.1 is a plot of the log-likelihood by iteration for the 100 runs of the fitting procedure in addition to histograms of other related quantities. The top left plot is the log-likelihood by iteration for all 100 runs. All runs have similar shape as seen when fitting the alignment model to simulated data (Section 9.2). The same gap estimates \( \hat{G}_1 = 2 \) and \( \hat{G}_2 = 11 \) were attained in all 100 runs and were attained in the first instance of updating the estimate of the gap positions. In all runs, the gap estimates did not change when updated a second time and so the fitting process halted. The top right plot of Figure 10.1 is a histogram of the total number of iterations for the 100 runs where we can see that on average, most runs converged in approximately 450 iterations, much less than the potential 10000.

In the bottom of Figure 10.1 we can see histograms of the initial and converged values of the log-likelihood for all 100 runs. As was the case when fitting the simulated data, the broad spread of the initial values of the log-likelihood compared to the converged values indicates that a broad range of initial estimates of the parameters all converged to values with very similar log-likelihood. Hence it is plausible that the maximum converged log-likelihood value is approximately a global maximum.

The estimated gap positions for the maximum log-likelihood run were \( \hat{G}_1 = 2 \) and \( \hat{G}_2 = 11 \) as shown in Figure 10.2. The estimated parameters \( \hat{\lambda} = \{\hat{a}, \hat{A}, \hat{B}\} \) that have the maximum log-

![Figure 10.2](image-url)
likelihood over all of the 100 runs are the Markov parameters

\[
\hat{a} = \begin{bmatrix}
0.0817 \\
0.2988 \\
0.3218 \\
0.2197 \\
0.0780
\end{bmatrix}, \quad \hat{A} = \begin{bmatrix}
0.7137 & 0.2411 & 0.0443 & 0.0003 & 0.0006 \\
0.1510 & 0.6995 & 0.1495 & 0.0001 & 0.0000 \\
0.0207 & 0.1615 & 0.6315 & 0.1768 & 0.0096 \\
0.0011 & 0.0092 & 0.1478 & 0.6300 & 0.2118 \\
0.0000 & 0.0000 & 0.0322 & 0.2615 & 0.7063
\end{bmatrix}
\] (10.1)

and the estimated parameters of the emission distributions

\[
X_t^{(i)} \mid \{S_t = j\} \sim \begin{cases} 
N(0.0225, 0.0010) & \text{if } j = 1 \\
N(0.1747, 0.0136) & \text{if } j = 2 \\
N(0.4674, 0.0390) & \text{if } j = 3 \\
N(0.7799, 0.0147) & \text{if } j = 4 \\
N(0.9436, 0.0030) & \text{if } j = 5.
\end{cases}
\] (10.2)

The estimated conditional emission densities are plotted in Figure 10.3. The states have been labeled 1 to 5 corresponding to the order of smallest to largest estimated means of the emission distributions. That is, left to right in Figure 10.3, the densities correspond to states 1 to 5 respectively. This labelling also applies to the Markov parameters (10.1).

![Figure 10.3: Densities corresponding to the estimated emission distributions (10.2).](image)
The estimated transition matrix $\hat{A}$ is approximately tridiagonal. The most likely outcome is to remain in the same state in a single time step and the further away the mean of the corresponding emission distribution, the less likely the state it is to be transitioned to in a single time step. Since a realised state sequence is more likely to transition between states with closer numbered labels, the corresponding observed emission sequence will most likely be consecutive observations from emission distributions with closer valued means. This suggests that highly erratic emission sequences are unlikely to be observed under the model.

The separation of the estimated means and the spread of the emission distributions suggests that different states broadly account for distinct levels of expression. However, note the particularly large variance of the emission distribution corresponding to state 3. We will show in Section 11.2 that this large variance arises because of the contamination of the estimated model parameters arising from the inclusion of pairs of profiles in the grapevine data for which the alignment under the $L(t)$-fold HMM is poor.

Figure 10.4: Heat map of the log-likelihood of the maximum likelihood estimated parameters $\hat{\lambda}$ given the grapevine data evaluated for each possible value of the gap positions.

Figure 10.4 is a heat map of the log-likelihood of the maximum likelihood estimated parameters $\hat{\lambda}$ evaluated for each possible value of the gap positions. The log-likelihood peaks at the estimated gap positions $\hat{G}_1 = 2$ and $\hat{G}_2 = 11$, and there is a decrease in log-likelihood as the values of the gap positions move further away from these values. This decrease is not monotonic nor as smooth...
as in the heat map obtained in the simulation example (Figure 9.1) and there a relatively sharp
decrease in log-likelihood for $G_1 \geq 11$ and $G_2 \leq 10$. However, the single peak structure indicates
that the gap positions appear to be reasonably well defined for the grapevine data.

10.2 Common Representations

Given the estimated $L(t)$-fold HMM, we find the most likely corresponding realisation of the state
sequence, the Viterbi path (Section 7.2), for each pair of aligned profiles and hence a common
representation of the expression profiles over the development cycle of the grape berries for each
gene. Example pairs of aligned expression profiles and corresponding Viterbi paths are plotted in
Figure 10.5. For the Viterbi paths, each state $j$ has been plotted at the estimated mean $\hat{\mu}_j$ of the
corresponding emission distribution given by (10.2), which makes the correspondence between the
Viterbi paths and the aligned expression profiles much clearer.

The Viterbi paths provide a simple representation of the gene expression from both vineyards
as they capture the information available in the pairs of observed profiles in a small number of
quantitative levels. The separation of the estimated means of the emission distributions suggests
that the Viterbi paths may be useful common representations of the aligned profiles. Two limita-
tions of the Viterbi paths are first that the state values and hence distinct levels of expression
are the same for all paths. Secondly, the Viterbi paths are bound by the arbitrary choice for the
number of states $N$ under the model. A different choice of $N$ will change the rendering of the
Viterbi paths as there will be more or less possible state values that the elements of the paths can
take.

Alternative common representations can be obtained by simply taking the average expression
level of the aligned profiles at each time point as given in Figure 10.6. The Viterbi paths are the
most obvious common representation as they are inherent under the model. However, given the
alignment of the expression profiles, it is possible to obtain other common representations such as
the average profiles. The aim is to combine the information from both the Willunga and Clare
vineyards into a single expression profile for each gene, which could then potentially be used for
further analysis. The collection of Viterbi paths or average profiles could be used as another data
set for cluster analysis for example, or for other analysis of interest as discussed for time course
microarray data in Chapter 1.
Figure 10.5: Aligned expression profiles from the Willunga (★) and Clare (★) vineyards. The elements of the Viterbi paths (○) have been plotted at the estimated means $\hat{\mu}_j$ of the corresponding emission distributions given by (10.2). Left to right, top to bottom, the pairs of aligned profiles correspond to genes with probe identification 1621649_at, 1610245_at, 1616418_at and 1609985_at.
Figure 10.6: Averaged expression profiles (⋆) between the Willunga and Clare vineyards after alignment. The elements of the Viterbi paths (○) have been plotted at the estimated means $\hat{\mu}_j$ of the corresponding emission distributions given by (10.2). Left to right, top to bottom, the averaged profiles correspond to genes with probe identification 1621649_at, 1610245_at, 1616418_at and 1609985_at.
Chapter 11

Alignment Diagnostics

For the pairs of aligned profiles plotted in Figure 10.5, the alignment and common representation appear to have worked well. The alignment procedure did not work as well for all pairs of profiles in the grapevine data. We know that there are pairs of profiles that are poorly aligned by our proposed method, either because they have:

1. Different shapes at the Willunga and Clare vineyards; or

2. The same basic shape but the alignment is not suitable.

Example pairs of profiles that correspond to these two issues can be seen in Figure 11.1. The pair of profiles on the left have different shape and so the alignment and Viterbi path common representation obtained are very poor. On the right, the pair of profiles have the same basic shape, an initial spike in expression level followed by low constant expression, however the alignment that was found for the grapevine data does not appear to have worked very well for this pair.

We investigate using the log-likelihood under the estimated $L(t)$-fold HMM to identify pairs of expression profiles that are poorly aligned. Ordering the pairs of profiles based on log-likelihood will enable this identification by the principle that poorly aligned profiles could be expected to have relatively low log-likelihood under the alignment model.

We obtain a log-likelihood ordering of the pairs of aligned profiles and show that the ordering correlates well with whether the pairs of profiles have been poorly aligned due to their different shapes (Section 11.1). However, the estimated model parameters are contaminated by the inclusion of the poorly aligned profiles. To address this problem, we additionally fit a more general HMM to the grapevine data and consider log-likelihood ordering using the estimated HMM parameters under the alignment $L(t)$-fold HMM (Section 11.2).
CHAPTER 11. ALIGNMENT DIAGNOSTICS

11.1 Ordering Based on Log-likelihood

We order the pairs of profiles based on the corresponding log-likelihood (8.5) calculated for the estimated $L(t)$-fold HMM parameters $\hat{\lambda}$ given by (10.1) and (10.2). To illustrate this ordering, selected pairs of profiles with rank ranging from 1 to 2062, the largest to smallest log-likelihood, are plotted in Figure 11.2.

There is a clear gradation of how well the pairs of profiles are aligned, starting at the top left and ending at the bottom right of Figure 11.2. The pair of profiles with rank 2062 plotted in the bottom right have different shape and for this reason the alignment and common representation attained are very poor. The low relative log-likelihood corresponding to this pair of aligned profiles is due to the large difference in observed expression levels at most time points. Since the expression levels at each time point are modelled by emission random variables with the same conditional distribution, pairs of observations become less likely as the distance between the observations increases.

The profiles in Figure 11.1 have rank, left to right, 2060 and 79 respectively. The rank of the pair of profiles on the left conforms to the broad result already described. The high rank of the pair of profiles on the right indicates that it has relatively high log-likelihood. This is the case since the alignment of this pair agrees very well at most time points except at the initial spike feature. The log-likelihood contribution from the time points where the alignment is poor are masked by the alignment at all other time points and the particularly large log-likelihood contribution of the small variance emission distribution corresponding to state 1, which can be inferred as so many
Figure 11.2: Aligned expression profiles from the Willunga (⋆) and Clare (⋆) vineyards. The elements of the Viterbi paths (○) have been plotted at the estimated means $\hat{\mu}_j$ of the corresponding emission distributions given by (10.2). Left to right, top to bottom, the pairs of aligned profiles correspond to genes with rank 1, 257, 515, 774, 1032, 1290, 1540, 1805 and 2062, and with probe identification 1612160_at, 1618211_at, 1619165_at, 1619345_s_at, 1612450_at, 1611324_at, 1613803_at, 1607255_at and 1607015_at.
elements of the corresponding Viterbi path are in state 1 (plotted at $\hat{\mu}_1 = 0.0225$).

11.2 Fitting an HMM and the Estimated HMM Parameters

Although the log-likelihood can be used to identify pairs of profiles that have different shape and so cannot be aligned, there are still pairs of profiles that are poorly aligned but have high relative log-likelihood. For the most part, the value of the Viterbi path at the time points with large difference in expression levels in Figure 11.1 corresponds to state 3 and the emission distribution with particularly large variance that we have flagged for further investigation (Section 10.1).

Under the $L(t)$-fold Baum-Welch algorithm, the observed pair of emissions at each time point have the same weighted contribution, $\zeta^{(k)}(j|\lambda^{[n-1]})$ given by (7.9), to the estimate of the emission distribution parameters. This can lead to particularly large estimated variances when the pairs of observed expression values are quite different, as seen at many time points in the poorly aligned profiles of Figure 11.1. The large estimated variance of the emission distribution corresponding to state 3 could be due to the contamination of the parameter estimates from the pairs of profiles that do not align well in the grapevine data.

One way to avoid such contamination is to consider a model where the expression profiles are not constrained to correspond to the same underlying state sequence. This could be achieved where the expression profiles from Willunga are modelled as the emission sequence of an HMM and the expression profiles from Clare are modelled as the emission sequence of an HMM where there are 2 time points at which no emission random variable is present, corresponding to the gap positions in the alignment model. However, we would not expect that fitting an HMM with gaps rather than just fitting a standard HMM to the profiles from the Clare vineyard would have any substantial effect on the estimate of the parameters.

Consider now that at the Willunga and Clare vineyards, each expression profile is modelled separately as the emission sequence of an HMM. We assume that the model parameters $\lambda$ are the same at both vineyards although due to the difference in length of the observed expression profiles, the length of the emission and state sequences will vary between models. However, the MLE of the common parameters can still be found using the Baum-Welch algorithm (Rabiner 1989).

We fit an $N = 5$ state HMM to the grapevine data. We ran the algorithm 100 times using the same initial parameter estimate $\lambda^{[0]}$ as in Section 10.1 in addition to the constraint that at the $n^{th}$ iteration, $(\sigma^2_j)^{[n]} \geq 0.001$ for $j = 1, 2, \ldots, 5$. The estimated HMM parameters with maximum log-likelihood over all runs, $\hat{\lambda}^* = \{\hat{a}^*, \hat{A}^*, \hat{B}^*\}$ are given below.
11.2.1 Comparison of Estimated Parameters

For the estimated HMM parameters \( \hat{\lambda}^* \), the states are also labeled 1 to 5 corresponding to the order of smallest to largest estimated means of the corresponding emission distributions. Roughly, the estimated HMM parameters \( \hat{\lambda}^* \) look very similar to the estimated \( L(t) \)-fold HMM parameters \( \hat{\lambda} \). First consider the estimated Markov parameters. We have

\[
\hat{a} = \begin{bmatrix}
0.0817 \\
0.2988 \\
0.3218 \\
0.2197 \\
0.0780
\end{bmatrix}, \quad \hat{\lambda} = \begin{bmatrix}
0.7137 & 0.2411 & 0.0443 & 0.0003 & 0.0006 \\
0.1510 & 0.6995 & 0.1495 & 0.0001 & 0.0000
\end{bmatrix}
\]

and

\[
\hat{a}^* = \begin{bmatrix}
0.1532 \\
0.2858 \\
0.2254 \\
0.2403 \\
0.0953
\end{bmatrix}, \quad \hat{\lambda}^* = \begin{bmatrix}
0.7032 & 0.2582 & 0.0268 & 0.0064 & 0.0054 \\
0.1898 & 0.6333 & 0.1531 & 0.0170 & 0.0069
\end{bmatrix}
\]

The estimated initial state probability vectors appear to be equivalent given their overall small contribution to the log-likelihood. The estimated transition matrices are both approximately tridiagonal and appear to have fairly similar elements.

<table>
<thead>
<tr>
<th>( \mu_1 )</th>
<th>( \sigma_1^2 )</th>
<th>( \mu_2 )</th>
<th>( \sigma_2^2 )</th>
<th>( \mu_3 )</th>
<th>( \sigma_3^2 )</th>
<th>( \mu_4 )</th>
<th>( \sigma_4^2 )</th>
<th>( \mu_5 )</th>
<th>( \sigma_5^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\lambda} )</td>
<td>0.0225</td>
<td>0.0010</td>
<td>0.1747</td>
<td>0.0136</td>
<td>0.4674</td>
<td>0.0390</td>
<td>0.7799</td>
<td>0.0147</td>
<td>0.9436</td>
</tr>
<tr>
<td>( \hat{\lambda}^* )</td>
<td>0.0233</td>
<td>0.0010</td>
<td>0.1973</td>
<td>0.0097</td>
<td>0.4784</td>
<td>0.0121</td>
<td>0.7720</td>
<td>0.0098</td>
<td>0.9553</td>
</tr>
</tbody>
</table>

Now consider the difference in the estimated parameters of the emission distributions. As seen in Table 11.1, the estimated variances are all smaller for the estimated HMM parameters \( \hat{\lambda}^* \) except for the estimate of \( \sigma_1^2 \), which in both cases is the minimum set value. The most apparent difference is in the variance of the emission distribution corresponding to state 3, which is clearly seen in Figure 11.3. We conclude that the two sets of estimated parameters \( \hat{\lambda} \) and \( \hat{\lambda}^* \) are very similar to each other and that the most apparent difference is the variance corresponding to state
CHAPTER 11. ALIGNMENT DIAGNOSTICS

3. This outcome conforms to the explanation of what can be expected when fitting an HMM to the grapevine data as detailed above.

11.2.2 Difference in Log-likelihood

We now consider ordering the pairs of profiles using the log-likelihood (8.5) of the estimated HMM parameters $\hat{\lambda}^*$ under the alignment $L(t)$-fold HMM with estimated gap positions $\hat{G}_1 = 2$ and $\hat{G}_2 = 11$. The scatter plot of the log-likelihood values $l(\hat{\lambda}|w_{1:19}^{(k)}, c_{1:17}^{(k)})$ and $l(\hat{\lambda}^*|w_{1:19}^{(k)}, c_{1:17}^{(k)})$ given each pair of expression profiles can be seen in Figure 11.4. For the most part, the log-likelihood values appear to be similar. We can see that we would obtain broad agreement between the log-likelihood orderings of the pairs of aligned profiles based on either set of estimated parameters.

In Figure 11.4 we see that $l(\hat{\lambda}|w_{1:19}^{(k)}, c_{1:17}^{(k)})$ is much greater than $l(\hat{\lambda}^*|w_{1:19}^{(k)}, c_{1:17}^{(k)})$ for all pairs of profiles with low log-likelihood. That is, the pairs of profiles previously identified as having different shape and for which the alignment is inappropriate (Section 11.1). A possible explanation for the difference in log-likelihood is that a large difference in expression levels at a given time is most likely under the emission distribution corresponding to state 3 and hence this emission distribution will have large weighted contribution to the log-likelihood at these time points. This can be inferred by considering that for the poorly aligned profiles, the corresponding Viterbi path is in state 3 for many time points when the observed expression levels are quite different (Section 11.1). Then when calculating the log-likelihood using $\hat{\lambda}^*$, the variance of the emission distribution corresponding to state 3 is much smaller and so a large reduction in log-likelihood can occur.

We can see in Figure 11.4 that there is also a small collection of pairs of aligned profiles
Figure 11.4: Scatter plot of the log-likelihood (8.5) of both estimated sets of parameters \( \hat{\lambda} \) and \( \hat{\lambda}^* \), and the equal log-likelihood line. The log-likelihood is calculated given each pair of expression profiles \( w_{1,19}^{(k)} \) and \( c_{1,17}^{(k)} \) for \( k = 1, 2, \ldots, 2062 \).

corresponding to high relative log-likelihood for which the same difference in log-likelihood is especially pronounced. Recall that the grapevine data is composed of 16 groups of genes that arise from prior cluster analysis that was performed on the profiles from the Clare vineyard (Chapter 1 and Appendix A). Figure 11.5 is the same log-likelihood plot as Figure 11.4 but where the log-likelihood corresponding to the profiles in Figures A.2 and A.5 are highlighted. These pairs of profiles account for most of the points that have relatively high log-likelihood but much lower log-likelihood when calculated using \( \hat{\lambda}^* \).

The profiles corresponding to Figures A.2 and A.5 all seem to have an initial spike in gene expression, followed by low constant expression. As discussed in Appendix A, it is the genes in these two groups in particular that appear to have more varied expression profiles at the Willunga vineyard. Example pairs of aligned profiles from these groups can be seen in Figure 11.6. Such pairs of aligned profiles generally have relatively high log-likelihood because they match very well for a large proportion of time points with low expression and hence have a large log-likelihood contribution from the small variance emission distribution corresponding to state 1 (Section 11.1). However, misalignment of the spike feature can lead to a drop in log-likelihood calculated using \( \hat{\lambda}^* \) as once again, the large difference in expression levels at a given time means a greater weighted log-likelihood contribution from the emission distribution corresponding to state 3.
When the spike features are well aligned, no such difference in log-likelihood is apparent. The aligned pair in the top left plot of Figure 11.6 corresponds to a difference in log-likelihood of approximately 8 and the alignment and common representation appear to have worked reasonably well. The remaining aligned profiles plotted in Figure 11.6 correspond to a difference in log-likelihood of more than 19 and it can be seen that the initial spike feature is misaligned in each case. Note that that there may be other pairs of aligned profiles in the grapevine data that have the same basic shape and high log-likelihood but where the alignment was not suitable using our proposed method. We have just used the profiles corresponding to those plotted in Figures A.2 and A.5 to broadly verify that a large difference in log-likelihood enables us to identify such pairs. To give an idea of the proportion of misaligned pairs, the profiles corresponding to Figures A.2 and A.5 make up 5.96% of the grapevine data, while the number of pairs with log-likelihood less than 0 calculated using $\hat{\lambda}$ is 5.82%.

Figure 11.5: Scatter plot of the log-likelihood (8.5) of both estimated sets of parameters $\hat{\lambda}$ and $\hat{\lambda}^*$. The log-likelihood values corresponding to the profiles in Figure A.2 are plotted in magenta (×), the log-likelihood values corresponding to the profiles in Figure A.5 are plotted in cyan (×).
Figure 11.6: Aligned expression profiles from the Willunga (•) and Clare (•) vineyards. The elements of the Viterbi paths (○) have been plotted at the estimated means $\hat{\mu}_j$ of the corresponding emission distributions given by (10.2). Left to right, top to bottom, the pairs of aligned profiles correspond to genes with probe identification 1620183_at, 1613056_at, 1622791_at and 1615196_at.
Chapter 12

Conclusion

We aimed to align the expression profiles in the grapevine data, obtain common representations for each pair of profiles and identify the pairs for which the proposed alignment was inappropriate (Section 1.2). We have achieved our stated aims in this project as we have:

1. Proposed a methodology for aligning the pairs of profiles from the Willunga and Clare vineyards based on HMMs;
2. Obtained common representations of each pair of aligned profiles under the model; and
3. Investigated using the proposed model to identify pairs of profiles for which the alignment is inappropriate.

In the following, we give a summary of the thesis and how we have achieved our aims (Section 12.1). We then consider some concluding discussion of our results and our proposed methodology, including key future work (Section 12.2).

12.1 Summary of Thesis

We reviewed HMMs (Chapter 3) and the mathematical solutions to the three associated problems (Chapter 4). We reviewed the application of Pair HMMs to the alignment problem for genomic sequence data and considered two ways in which these models could be extended to model time course microarray data (Chapter 5). We reviewed a number of papers in which the authors proposed models that could be considered as one of these identified extensions as well as other HMM based alignment methodologies that we determined were inappropriate for our purposes.
We proposed an extension to HMMs, $L(t)$-fold HMMs (Chapter 6), where there are $L(t)$ emission random variables at each time point $t$. We adapted the existing HMMs solutions to solve the analogous associated problems for $L(t)$-fold HMMs (Chapter 7). We showed how an $L(t)$-fold HMM enables the alignment of different length sequences and proposed an alignment model for the grapevine data (Chapter 8). We presented a methodology for fitting the alignment model, which we demonstrated on simulated data (Chapter 9).

We implemented our proposed alignment methodology for the grapevine data and gave an interpretation of the output (Chapter 10). For each aligned pair, we found the Viterbi path, the most likely corresponding realisation of the state sequence under the model. The Viterbi paths are one possible common representation of the aligned profiles from the Willunga and Clare vineyards over the development of the grape berries for each gene. We investigated alignment diagnostics under the model to identify pairs of expression profiles for which the proposed alignment was inappropriate (Chapter 11). We presented and evaluated a methodology based on the log-likelihood of both estimated $L(t)$-fold HMM and HMM parameters under the proposed alignment $L(t)$-fold HMM for this task.

12.2 Discussion

There are a number of issues with the alignment $L(t)$-fold HMM that could be addressed in future work. As previously discussed, emission random variables at the same time point can be swapped between the defined emission sequences and yet the model will retain the same distribution. This could be addressed by adding correlation between consecutive emission random variables in the same sequence. Also, the conditional emission densities are Gaussian while the range of the grapevine data is only the interval $[0, 1]$ with a large proportion of expression levels equal to 0 or 1. It is possible that the emission distributions could be altered in order to better model the distribution of the expression levels. Although the choice of $N = 5$ states in our proposed model appears suitable, the correct choice of the number of states in an HMM is similar to the problem of choosing the number of mixture components in a mixture model or the number of clusters for cluster analysis. Many methods have been proposed to tackle such problems, however no standard methodology exists (Rabiner 1989).

Notwithstanding, we have shown that the alignment methodology developed here works well and have obtained good results when fitting the model to the grapevine data. The estimated model parameters have sensible interpretations and the estimated gap positions appear to have been
reasonably well determined. The alignment could be evaluated by comparing it to the alignment obtained using other methodologies such as given by Aach & Church (2001). The obtained common representations, either the Viterbi paths or average profiles, could readily be used for further analysis such as clustering, or for other analysis as discussed in Chapter 1. It may also be useful to investigate implementing the alignment procedure before other stages of the data pre-processing as described in Appendix A for better results.

For simplicity in this project we only considered data from the Willunga and Clare vineyards in the original time course experiment. The proposed alignment methodology could be extended to accommodate more than two vineyards and hence expression profiles for each gene as to begin with, more than two emission sequences are easily defined under an $L(t)$-fold HMM. Additionally, the alignment diagnostics under the model could be further investigated as this aspect of our methodology is particularly useful in practice. We presented a methodology based on log-likelihood to identify pair of profiles with poor alignment under the model and further work could be done towards automating this process. For instance, we broadly verified that the identification works but did not consider a threshold or cut-off point to split the data into well and poorly aligned subsets.
Appendices
Appendix A

The Grapevine Data

The motivating data set for this project is composed of measurements from a time course microarray experiment using Affymetrix GeneChip arrays that was conducted by Dr Chris Davies, Senior Research Scientist, CSIRO Plant Industry Research Division. The experiment was conducted on grapevines at a number of different vineyards in South Australia and was carried out over the development cycle of the grape berries.

The species and variety of grapevine, *Vitis vinifera* L. and Cabernet Sauvignon respectively, were the same at all vineyards and the individual grapevine plants are considered biologically identical at each vineyard. There were different environmental conditions between vineyards due to the different locations and different years (grape growing seasons) over which the experiment took place. In addition, there were different treatment conditions within vineyards, the grapevines being subject to either spur pruning or machine pruning.

The levels of gene expression were measured weekly over the development cycle of the grape berries, which is the closed-flower to ripe-red stages of the berries themselves. The decision to start and stop the time course was made by the field researchers collecting the data and was based on the observed development of the grape berries at the particular vineyard concerned. At each time point, the grapevine material sampled for input into the microarray was considered to be representative of the berry development over the entire vineyard.

We focus on the observed expression profiles from two of the vineyards that are situated in the Willunga and Clare grape growing regions of South Australia. In addition to being from spatially distinct vineyards (Figure 1.2), the data was collected in the 2004 grape growing season at Willunga and in the 2005 grape growing season at Clare. The length of the development cycle was 19 weeks.
at the Willunga vineyard and 17 weeks at the Clare vineyard.

There were 2 replicate expression measurements at each time point at the Willunga vineyard where the grapevines were only subject to spur pruning. There were 2 replicate expression measurements at each time point for each of the two treatments, spur and machine pruning, at the Clare vineyard. From prior analysis (Chris Davies, personal communication), the different pruning methods were found to have no effect on the gene expression levels and so we consider that we have 4 replicates at each time point at the Clare vineyard.

We obtain a single profile for each gene at the Willunga and Clare vineyards by averaging the replicate gene expression measurements at each time point. Although there are more replicates at the Clare vineyard, the component of variance that is decreased by taking this average is the variance associated with the microarray technology. There is still variance associated with the grapevine plants or across the vineyard for example, so we do not take the different number of replicates into account when modelling the averaged profiles.

In this project we are not interested in amplitude of expression level but rather in the changes in gene expression over time and hence the shape of the expression profiles. For this reason, we scale the data so that each observed expression level lies between 0 and 1. For each expression profile, we transform the observation at each time point by subtracting the minimum observed expression level and then dividing by the maximum observation minus the minimum. Note that the distinct maximum and minimum expression levels are found for each profile individually.

Dr Davies selected 2062 genes that are thought to be associated with the development cycle of the grape berries for use in this project. It is expected that genes associated with the development cycle of the grape berries will exhibit similar gene expression at each vineyard and hence the corresponding pairs of profiles will be conducive to an alignment. Note that all of the genes we are considering in this project were found to exhibit significant differential expression (Chris Davies, personal correspondence). Cluster analysis had been carried out on the expression profiles from the Clare vineyard and the collection of genes selected for this project correspond to 16 of the groups from that analysis. We refer to the collection of scaled average expression profiles from the Willunga and Clare vineyards that correspond to these 2062 genes as the ‘grapevine data’.

The clustering procedure used was $k$-medoids clustering with the Euclidean distance. Note that $k$-medoids clustering is a similar procedure to $k$-means but where the cluster centres are observed values rather than a cluster average (Hastie, Tibshirani & Friedman 2009). We can see that for all 16 groups, the profiles from the Clare vineyard appear to have the same basic shape within cluster
(Figures A.1 to A.16). The groups of the Willunga profiles appear to generally adhere to the same corresponding shape, except with more variability at each time point than the profiles from Clare, which the clustering was based on. Notably, the profiles from Willunga in Figures A.2 and A.5 have much greater variability compared to the corresponding Clare profiles, to the extent that it is not obvious whether most of the Willunga profiles all share the same basic shape.
Figure A.1: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 96 genes in group 1.
Figure A.2: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 71 genes in group 2.
Figure A.3: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 115 genes in group 3.
Figure A.4: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 115 genes in group 4.
Figure A.5: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 52 genes in group 5.
Figure A.6: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 193 genes in group 6.
Figure A.7: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 153 genes in group 7.
Figure A.8: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 178 genes in group 8.
Figure A.9: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 149 genes in group 9.
Figure A.10: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 140 genes in group 10.
Figure A.11: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 147 genes in group 11.
Figure A.12: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 135 genes in group 12.
Figure A.13: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 167 genes in group 13.
Figure A.14: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 87 genes in group 14.
Figure A.15: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 180 genes in group 15.
Figure A.16: Expression profiles from the Willunga (top) and Clare (bottom) vineyards for the 84 genes in group 16.
Appendix B

The EM Algorithm

The EM algorithm is used to find the MLE of model parameters where there is incomplete or missing data, or in which additional missing data is suitably specified. A key reference for the general EM algorithm is the unifying paper of Dempster, Laird & Rubin (1977), however the EM algorithm as applied to the HMM MLE problem is know as the Baum-Welch algorithm as it was proposed prior to the general EM form by Baum et al. (1970). Here we present the mathematical results necessary to define the general EM algorithm. The following review is primarily based on the work of McLauchlan & Krishnan (1997) and Bilmes (1998).

Consider random variables $Y$ and $Z$ with known joint density function $p(y, z|\theta)$ and parameter(s) $\theta$. Given observed data $y$, but missing data $z$, we wish to make inference about $\theta$. This can be achieved by considering the marginal log-likelihood of the parameters given the observed data. The maximum likelihood estimator of $\theta$ is given by

$$\hat{\theta} = \arg\max_\theta l(\theta|y).$$

In many cases under the maximum likelihood framework, it is not possible to obtain an analytic expression for $\hat{\theta}$ nor is it straightforward to numerically maximise the log-likelihood. The EM algorithm is a procedure for iteratively maximising the observed data log-likelihood based on the consideration of the joint density of the observed and missing data. The joint density of the observed and missing data (full data) must be known and such that the full data log-likelihood can be maximised with respect to $\theta$. 
Algorithm B.1 The EM Algorithm.

1. Obtain initial estimate of the parameters, \( \theta^{[0]} \).

2. **Expectation step:** At the \( n \)th iteration, calculate

\[
Q(\theta|\theta^{[n-1]}) = E_Z \left[ l(\theta|y, z) \bigg| y, \theta^{[n-1]} \right].
\]

3. **Maximisation step:** Find

\[
\theta^{[n]} = \operatorname{argmax}_\theta Q(\theta|\theta^{[n-1]}).
\]

4. Return to the Expectation step for a specified number of iterations or until some convergence criterion is satisfied.

That the observed data log-likelihood of the update estimate \( \theta^{[n]} \) is greater than or equal to than the observed data log-likelihood of the current estimate \( \theta^{[n-1]} \) is given in the following result.

**Theorem B.1 EM Update.** Given current estimate \( \theta^{[n-1]} \),

\[
l(\theta^{[n]}|y) \geq l(\theta^{[n-1]}|y)
\]

where

\[
\theta^{[n]} = \operatorname{argmax}_\theta Q(\theta|\theta^{[n-1]}).
\]

**Proof of Theorem B.1.** A proof is given by McLauchlan & Krishnan (1997). \( \square \)

Since the observed data log-likelihood values are monotonic non-decreasing, if the log-likelihood surface is bounded above, \( l(\theta|y) < M \forall \theta \), then the sequence of log-likelihood values

\[
l(\theta^{[0]}|y), l(\theta^{[1]}|y), l(\theta^{[2]}|y), \ldots
\]

converges. Under fairly general conditions, the log-likelihood values converge to a stationary point (McLauchlan & Krishnan 1997). However, the sequence (B.1) is not guaranteed to converge to the global maximum of the log-likelihood and depending on the complexity of the log-likelihood
surface and initial estimate of the parameters, may only converge to a local maximum or stationary point.

To account for the effect of the initial parameter estimate on the log-likelihood convergence, the EM algorithm is run as many times as possible in practice with different initial parameter estimate \( \theta^{(0)} \). The initial estimate is chosen so that over multiple runs of the EM algorithm, the initial estimate covers the domain of the parameters, in addition to incorporating any other information for the particular data set and application.

Generally the convergence criterion for stopping the EM algorithm is a minimum increase in the log-likelihood or a minimum change in the parameter estimates at each iteration. The rate of convergence of the EM algorithm is widely held to be slow, which has lead to the development of a number of methods to speed up convergence that are not discussed here. Rates of convergence are treated by McLauchlan & Krishnan (1997), as are many other results and extensions for the EM algorithm.
Appendix C

Proof of Theorem 4.6

We maximise (4.16) with respect to $\lambda = \{a, A, B\}$. We first rewrite (4.16) to be linear in functions of disjoint subsets of the parameters $\lambda$. Hence we maximise each function with respect to the corresponding parameters individually.

Define indicator functions

$$I_t^{(k)}(j) = \begin{cases} 1 & \text{if } S_t^{(k)} = j \\ 0 & \text{otherwise} \end{cases}$$

and

$$J_t^{(k)}(i,j) = \begin{cases} 1 & \text{if } S_{t-1}^{(k)} = i \text{ and } S_t^{(k)} = j \\ 0 & \text{otherwise}. \end{cases}$$

Rewrite the Markov parameters

$$a_{s_t^{(k)}} = \prod_{i=1}^{N} a_t^{(i)} I_t^{(k)}(i),$$

$$a_{s_{t-1}^{(k)}, s_t^{(k)}} = \prod_{i=1}^{N} a_{t-1}^{(i,j)} J_t^{(k)}(i,j)$$

and the conditional emission densities

$$b(x_t^{(k)}|\mu_{s_t^{(k)}}, \sigma_{s_t^{(k)}}^2) = \prod_{j=1}^{N} \left( b(x_t^{(k)}|\mu_j, \sigma_j^2) \right)^{I_t^{(k)}(j)}. $$
Hence rewrite the joint density

\[
p(x^{(1:K)}_{1:T}, s^{(1:K)}_{1:T} | \lambda) = \prod_{k=1}^{K} \left[ \prod_{t=2}^{T} a_{s^t_{i_k} s^t_{i_{t-1}}} \left( \prod_{t=1}^{T} b(x^{(k)}_t | \mu_{s^t_{i_k}}, \sigma^2_{s^t_{i_k}}) \right) \right]
\]

\[
= \prod_{k=1}^{K} \left[ \left( \prod_{i=1}^{N} a_{I_{i}^{(k,i)}} \right) \left( \prod_{t=2}^{T} \prod_{i=1}^{N} a_{I_{ij}^{(k,i,j)}} \right) \left( \prod_{t=1}^{T} \prod_{j=1}^{N} b(x^{(k)}_t | \mu_{j}, \sigma^2_{j}) \right) \right]
\]

and the full data log-likelihood

\[
l(\lambda | x^{(1:K)}_{1:T}, s^{(1:K)}_{1:T}) = \sum_{k=1}^{K} \sum_{i=1}^{N} I_{i}^{(k,i)} \log[a_i] + \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{ij}^{(k,i,j)} \log[a_{ij}] + \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{j=1}^{N} I_{ij}^{(k,j)} \log[b(x^{(k)}_t | \mu_{j}, \sigma^2_{j})]. \quad (C.1)
\]

Now

\[
E_{S^{(1:K)}_{1:T}} \left[ I_{ij}^{(k,j)} | x^{(1:K)}_{1:T}, \lambda^{[n-1]} \right] = p(S^{(k)}_t = j | x^{(1:K)}_{1:T}, \lambda^{[n-1]})
\]

\[
= p(S^{(k)}_t = j | x^{(k)}_t, \lambda^{[n-1]})
\]

\[
= \zeta^{(k)}_t (j | \lambda^{[n-1]}). \quad (C.2)
\]

The second line follows as \( S^{(k)}_{t-1}, S^{(k)}_t \perp \perp X^{(t-1)}_{1:T} \) from (4.9). Additionally

\[
E_{S^{(1:K)}_{1:T}} \left[ J_{ij}^{(k,i,j)} | x^{(1:K)}_{1:T}, \lambda^{[n-1]} \right] = p(S^{(k)}_{t-1} = i, S^{(k)}_t = j | x^{(1:K)}_{1:T}, \lambda^{[n-1]})
\]

\[
= p(S^{(k)}_{t-1} = i, S^{(k)}_t = j | x^{(k)}_t, \lambda^{[n-1]})
\]

\[
= \xi^{(k)}_t (i, j | \lambda^{[n-1]}). \quad (C.3)
\]

The second line follows as \( \{S^{(k)}_{t-1}, S^{(k)}_t \} \perp \perp X^{(t-1)}_{1:T} \) from (4.9).
Hence

\[ Q(\lambda|\lambda^{[n-1]}) = E_{S_1^{(1,K)}} \left[ I(\lambda|x_1^{(1,K)}, q_1^{(1,K)}) \middle| x_1^{(1,K)}, \lambda^{[n-1]} \right] \]

\[ = \sum_{k=1}^{K} \sum_{i=1}^{N} E_{S_1^{(1,K)}} \left[ I_1^{(k)}(i) \middle| x_1^{(1,K)}, \lambda^{[n-1]} \right] \log[a_i] \]

\[ + \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} E_{S_1^{(1,K)}} \left[ J_t^{(k)}(i,j) \middle| x_1^{(1,K)}, \lambda^{[n-1]} \right] \log[a_{ij}] \]

\[ + \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{j=1}^{N} E_{S_1^{(1,K)}} \left[ I_t^{(k)}(j) \middle| x_1^{(1,K)}, \lambda^{[n-1]} \right] \log[b(x_t^{(k)} | \mu_j, \sigma^2_j)] \]

\[ = \sum_{k=1}^{K} \sum_{i=1}^{N} \xi_t^{(k)}(i|\lambda^{[n-1]}) \log[a_i] + \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi_t^{(k)}(i,j|\lambda^{[n-1]}) \log[a_{ij}] \]

\[ + \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{j=1}^{N} \xi_t^{(k)}(j|\lambda^{[n-1]}) \log[b(x_t^{(k)} | \mu_j, \sigma^2_j)]. \]

(C.4)

The second line follows from (C.1). The third line follows from (C.2) and (C.3). Note that the weights \( \xi_t^{(k)}(j|\lambda^{[n-1]}) \) and \( \xi_t^{(k)}(i,j|\lambda^{[n-1]}) \) are evaluated using the current estimate of the parameters \( \lambda^{[n-1]} \) and so are constant with respect to \( \lambda \). Hence, (C.4) is linear in functions of disjoint subsets \( a, A \) and \( B \) of the parameters \( \lambda \). We maximise each of these functions with respect to the corresponding parameters individually.

First we maximise

\[ \sum_{k=1}^{K} \sum_{i=1}^{N} \xi_t^{(k)}(i|\lambda^{[n-1]}) \log[a_i] \]

with respect to \( a = (a_1, a_2, \ldots, a_N)^T \) such that \( \sum_{i=1}^{N} a_i = 1 \) (Definition 3.2). Set up the Lagrangian

\[ L(a, c) = \sum_{k=1}^{K} \sum_{i=1}^{N} \xi_t^{(k)}(i|\lambda^{[n-1]}) \log[a_i] - c \left( \sum_{i=1}^{N} a_i - 1 \right) \]

where \( c \) is the Lagrange constant. For a stationary point of (C.5) we will have \( \frac{\partial L}{\partial a_i} = 0 \) for \( i = 1, 2, \ldots, N \). Hence

\[ \frac{\partial L}{\partial a_i} = \sum_{k=1}^{K} \xi_t^{(k)}(i|\lambda^{[n-1]}) \cdot \frac{1}{a_i} - c \]

\[ \Rightarrow \quad a_i = \frac{1}{c} \sum_{k=1}^{K} \xi_t^{(k)}(i|\lambda^{[n-1]}) \]

for \( i = 1, 2, \ldots, N \) is a stationary point.
Now
\[\sum_{i=1}^{N} a_i = 1 \quad \text{and} \quad \sum_{i=1}^{N} a_i = \sum_{i=1}^{N} \frac{1}{c} \sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]}) \Rightarrow 1 = \sum_{i=1}^{N} \frac{1}{c} \sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]}) \]
\[\Rightarrow c = \sum_{i=1}^{N} \sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]}) = K \]

since \(\sum_{i=1}^{N} \zeta_1^{(k)}(i|\lambda^{[n-1]}) = 1\) by definition (4.14). Hence \(a_\lbrack n \rbrack\) where
\[a_\lbrack n \rbrack_i = \frac{1}{K} \sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]})\]
for \(i = 1, 2, \ldots, N\) is a stationary point of (C.5).

Now
\[
\frac{\partial^2}{\partial a_i^2} \left( \sum_{k=1}^{K} \sum_{i=1}^{N} \zeta_1^{(k)}(i|\lambda^{[n-1]}) \log[a_i] \right) = -\frac{\sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]})}{a_i^2}
\]
for \(i = 1, 2, \ldots, N\) and

\[
\frac{\partial^2}{\partial a_i \partial a_{i'}} \left( \sum_{k=1}^{K} \sum_{i=1}^{N} \zeta_1^{(k)}(i|\lambda^{[n-1]}) \log[a_i] \right) = 0
\]
for \(i, i' = 1, 2, \ldots, N\) such that \(i \neq i'\). Since both \(\zeta_1^{(k)}(i|\lambda^{[n-1]}) \geq 0\) and \(a_i \geq 0\), and \(a_\lbrack n \rbrack_i = 0\) if and only if \(\sum_{k=1}^{K} \zeta_1^{(k)}(i|\lambda^{[n-1]}) = 0\), we have
\[
\left. \frac{\partial^2}{\partial a_i} \left( \sum_{k=1}^{K} \sum_{i=1}^{N} \zeta_1^{(k)}(i|\lambda^{[n-1]}) \log[a_i] \right) \right|_{a_i = a_\lbrack n \rbrack_i} \leq 0
\]
for \(i = 1, 2, \ldots, N\). Hence the Hessian matrix of (C.5) with respect to \(a\) is negative definite at \(a_\lbrack n \rbrack\) and so \(a_\lbrack n \rbrack\) is a local maximum of (C.5).

Now we maximise
\[\sum_{k=1}^{K} \sum_{i=2}^{T} \sum_{j=1}^{N} \sum_{i=1}^{N} \zeta_1^{(k)}(i, j|\lambda^{[n-1]}) \log[a_{ij}] \quad (\text{C.6})\]
with respect to \(A = \{a_{ij}\}\) such that \(\sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} = N\) (Definition 3.2). The Lagrangian is
\[
L(A, c) = \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) \log[a_{ij}] - c \left( \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} - N \right)
\]

and \(\frac{\partial L}{\partial a_{ij}} = 0\) for \(i, j = 1, 2, \ldots, N\) at a stationary point of (C.6). Hence

\[
\frac{\partial L}{\partial a_{ij}} = \sum_{k=1}^{K} \sum_{t=2}^{T} \frac{\xi_{t}^{(k)}(i, j|\lambda^{[n-1]})}{a_{ij}} - c
\]

\[
\Rightarrow \quad a_{ij} = \frac{1}{c} \sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})
\]

for \(i, j = 1, 2, \ldots, N\) at a stationary point.

Now

\[
\sum_{j=1}^{N} a_{ij} = 1 \quad \text{and} \quad \sum_{j=1}^{N} a_{ij} = \frac{1}{c} \sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})
\]

\[
\Rightarrow \quad 1 = \frac{1}{c} \sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})
\]

\[
\Rightarrow \quad c = \sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})
\]

\[
= \sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i|\lambda^{[n-1]})
\]

since \(\sum_{j=1}^{N} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) = \xi_{t-1}^{(k)}(i|\lambda^{[n-1]})\) by definition (4.15). Hence \(A^{[n]}\) where

\[
a_{ij}^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})}{\sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i|\lambda^{[n-1]})}
\]

for \(i, j = 1, 2, \ldots, N\) is a stationary point of (C.6).

Now

\[
\frac{\partial^{2}}{\partial a_{ij}^{2}} \left( \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) \log[a_{ij}] \right) = \frac{\sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]})}{a_{ij}^{2}}
\]

for \(i, j = 1, 2, \ldots, N\) and

\[
\frac{\partial^{2}}{\partial a_{ij} \partial a_{i'j'}} \left( \sum_{k=1}^{K} \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) \log[a_{ij}] \right) = 0
\]

for \(i, j, i', j' = 1, 2, \ldots, N\) such that it is not that case that \(i = i'\) and \(j = j'\) at the same time. Since \(\xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) \geq 0\) and \(a_{ij} \geq 0\), and \(a_{ij}^{[n]} = 0\) if and only if \(\sum_{k=1}^{K} \sum_{t=2}^{T} \xi_{t}^{(k)}(i, j|\lambda^{[n-1]}) = 0\),
we have
\[ \frac{\partial^2}{\partial a_{ij}^2} \left( \sum_{k=1}^K \sum_{l=2}^T \sum_{t=1}^N \sum_{j=1}^N \xi_t^{(k)}(i, j|\lambda^{[n-1]}) \log[a_{ij}] \right) \bigg|_{a_{ij}=a_{ij}^{[n]}} \leq 0 \]
for \( i, j = 1, 2, \ldots, N \). Hence the Hessian matrix of (C.6) is negative definite at \( A^{[n]} \) and so \( A^{[n]} \) is a local maximum of (C.6).

Finally, we maximise
\[ \sum_{k=1}^K \sum_{t=1}^T \sum_{j=1}^N \xi_t^{(k)}(j|\lambda^{[n-1]}) \log[b(x_t^{(k)}|\mu_j, \sigma_j^2)] \quad (C.7) \]
with respect to \( B = \{ \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \ldots, \mu_N, \sigma_N^2 \} \). Recall
\[ b(x|\mu_j, \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left\{ -\frac{1}{2\sigma_j^2} (x - \mu_j)^2 \right\}. \]
Hence we maximise
\[ \phi(B) = \sum_{k=1}^K \sum_{t=1}^T \sum_{j=1}^N \xi_t^{(k)}(j|\lambda^{[n-1]}) \left[ -\frac{1}{2} \log[2\pi\sigma_j^2] - \frac{1}{2\sigma_j^2} (x_t^{(k)} - \mu_j)^2 \right] \]
with respect to \( B \). We will first maximise \( \phi \) with respect to \( \mu_j \) for \( j = 1, 2, \ldots, N \) simultaneously and then maximise \( \phi \) with respect to \( \sigma_j^2 \) for \( j = 1, 2, \ldots, N \) simultaneously.

\[ \frac{\partial \phi}{\partial \mu_j} = \sum_{k=1}^K \sum_{t=1}^T \xi_t^{(k)}(j|\lambda^{[n-1]}) \frac{1}{2\sigma_j^2} (x_t^{(k)} - \mu_j) \]
and
\[ \frac{\partial \phi}{\partial \mu_j} = 0 \quad \Rightarrow \quad \mu_j^{[n]} = \frac{\sum_{k=1}^K \sum_{t=1}^T \xi_t^{(k)}(j|\lambda^{[n-1]}_t) x_t^{(k)}}{\sum_{k=1}^K \sum_{t=1}^T \xi_t^{(k)}(j|\lambda^{[n-1]})} \]
for \( j = 1, 2, \ldots, N \) is a stationary point of (C.7).

Now
\[ \frac{\partial^2 \phi}{\partial \mu_j^2} = \sum_{k=1}^K \sum_{t=1}^T \frac{\xi_t^{(k)}(j|\lambda^{[n-1]})}{2\sigma_j^2} \]
and

\[ \frac{\partial^2 \phi}{\partial \mu_j \partial \mu_{j'}} = 0 \]

for \( j, j' = 1, 2, \ldots, N \) such that \( j \neq j' \). Since \( \zeta^{(k)}_t(j|\lambda^{[n-1]}) \geq 0 \) and \( \sigma^2_j > 0 \) for \( t = 1, 2, \ldots, T \) and \( j = 1, 2, \ldots, N \),

\[ \frac{\partial^2 \phi}{\partial \mu_j^2} < 0 \]

for \( j = 1, 2, \ldots, N \) and so the Hessian matrix of \( \phi \) with respect to \( \mu_j \) is negative definite at \( \mu_j^{[n]} \) for \( j = 1, 2, \ldots, N \). Hence the stationary point \( \mu_j^{[n]} \) is a local maximum of (C.7) for \( j = 1, 2, \ldots, N \).

Now consider maximising \( \phi \) with respect to \( \sigma^2_j \) for \( j = 1, 2, \ldots, N \).

\[ \frac{\partial \phi}{\partial \sigma^2_j} = \sum_{k=1}^{K} \sum_{t=1}^{T} \zeta^{(k)}_t(j|\lambda^{[n-1]}) \left[ -\frac{1}{2\sigma^2_j} + \frac{1}{(\sigma^2_j)^2} (x^{(k)}_t - \mu^{[n]}_j)^2 \right] \]

and so

\[ \frac{\partial \phi}{\partial \sigma^2_j} = 0 \quad \Rightarrow \quad (\sigma^2_j)^{[n]} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta^{(k)}_t(j|\lambda^{[n-1]}) (x^{(k)}_t - \mu^{[n]}_j)^2}{\sum_{k=1}^{K} \sum_{t=1}^{T} \zeta^{(k)}_t(j|\lambda^{[n-1]})} \]

for \( j = 1, 2, \ldots, N \) is a stationary point of (C.7).

Now

\[ \frac{\partial^2 \phi}{\partial (\sigma^2_j)^2} = \sum_{k=1}^{K} \sum_{t=1}^{T} \zeta^{(k)}_t(j|\lambda^{[n-1]}) \left[ \frac{1}{2(\sigma^2_j)^2} - \frac{1}{(\sigma^2_j)^3} (x^{(k)}_t - \mu^{[n]}_j)^2 \right] \]

and

\[ \frac{\partial^2 \phi}{\partial \sigma^2_j \partial \sigma^2_{j'}} = 0 \]

for \( j, j' = 1, 2, \ldots, N \) such that \( j \neq j' \).
Consider

\[
\frac{\partial^2 \phi}{\partial (\sigma_j^2)^2} < 0 \iff \sum_{k=1}^K \sum_{t=1}^T \zeta_t^{(k)}(j|\ni) \left[ \frac{1}{2(\sigma_j^2)^2} - \frac{1}{(\sigma_j^2)^3} (\mu_j^{[n]} - \mu_j^{[n]} - \mu_j^{[n]})^2 \right] < 0
\]

\[
\iff \sum_{k=1}^K \sum_{t=1}^T \zeta_t^{(k)}(j|\ni) \frac{1}{2(\sigma_j^2)^2} < \sum_{k=1}^K \sum_{t=1}^T \zeta_t^{(k)}(j|\ni) \frac{1}{(\sigma_j^2)^3} (\mu_j^{[n]} - \mu_j^{[n]})^2
\]

\[
\iff \sigma_j^2 < \frac{2(\sum_{k=1}^K \sum_{t=1}^T \zeta_t^{(k)}(j|\ni) (x_t^{(k)} - \mu_j^{[n]}))^2}{\sum_{k=1}^K \sum_{t=1}^T \zeta_t^{(k)}(j|\ni)}
\]

\[
\implies \sigma_j^2 < 2(\sigma_j^2)^{[n]}
\]

This is true at \((\sigma_j^2)^{[n]}\) for \(j = 1, 2, \ldots, N\). Hence the Hessian matrix of \(\phi\) with respect to \(\sigma_j^2\) for \(j = 1, 2, \ldots, N\) is negative definite at \((\sigma_j^2)^{[n]}\) and so \((\sigma_j^2)^{[n]}\) is a local maximum of (C.7) for \(j = 1, 2, \ldots, N\). □
Bibliography


Murphy, K. (2005), ‘The HMM Toolbox’.  


