UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Modeling of Biochemical States of DNA Replication Using Hidden Markov Models

Permalink

https://escholarship.org/uc/item/7dp9d72q

Author

Simms, Matthew McCracken

Publication Date

2018

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

MODELING OF BIOCHEMICAL STATES OF DNA REPLICATION USING HIDDEN MARKOV MODELS

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

DOCTOR OF PHILOSOPHY

in

APPLIED MATHEMATICS AND STATISTICS

by

Matthew Simms

March 2019

The Dissertation of Matthew Simms
is approved:

Hongyun Wang, Co-Chair

Herbert Lee, Co-Chair

Qi Gong

Mark Akeson

Lori Kletzer
Vice Provost and Dean of Graduate Studies

Copyright © by

Matthew Simms

2019

Contents

List of Figures

List of Tables

Abstract

Modeling of Biochemical States of DNA Replication Using Hidden Markov Models

by

Matthew Simms

In nanopore experiments, DNA replication facilitated by ϕ29 DNA polymerase (DNAP) can be observed at the single molecule level. The biochemical state of the DNA-DNAP complex was studied by setting the complex atop a α-hemolysin nanopore and applying an electric voltage. The movement of the DNA strand relative to the nanopore was observed on a single base pair level by the ionic current blockade. The time trace of the recorded ionic current amplitude from these experiments was used to study the biochemical states. Given that the recorded amplitude of the ionic current was an indirect measurement of the true amplitude level, which in turn was an indirect measurement of the true biochemical state, the experiments were modeled as a Hidden Markov Chain (HMC). When the DNA position of two biochemical states relative to the nanopore is the same, the states yield the same current amplitude level. To extract the dynamic transition rates between biochemical states that are not distinguishable in amplitude level, two methodologies were applied to study the HMC. The first was a fully Bayesian model, for which Markov chain Monte-Carlo (MCMC) simulations were used to infer the reaction rates in a system of three biochemical states with two observed amplitude levels. The second model adopted concepts of Viterbi training or the segmental k-means algorithm to find point estimates of the transition rates. Given the low transition probabilities, the properties of the second model led to a substantial bias in inference. The bias was addressed by first using a meta-model to describe the relation between the generating transition rates and the biased inference. Then the inverse problem of the meta-model was solved to reduce the bias in the inference. The meta-model was a fully Bayesian Gaussian process model, built by creating a series of computer-generated datasets around the given dataset. Improved inference was obtained by drawing posterior samples from the inverse of the meta-model. Since both the MCMC simulations and bias reduction techniques resulted in simulated posteriors, the two methods were used to confirm each other. In the comparison of these two methods, the approach of the second model plus bias reduction has the advantage of achieving similar inference accuracy at a much lower computational cost.

For my wife and children, Sarah, Cora, and Henry

Acknowledgments

This dissertation was the culmination of years of research and would not have been possible without significant support. First, I would like to thank the members of the Reading Committee. Reading a technical paper is a time-consuming endeavor, and I appreciate the committee taking time to read the document, attend the defense, and provide thoughtful input. Among the committee's members, my advisors Dr. Herbert Lee and Dr. Hongyun Wang deserve special recognition. Both Herbie and Hongyun were extremely patient, encouraging, and supportive of my work. In addition, both allowed the research to take its own direction, regardless of the topic area. In particular, I appreciate Hongyun for always encouraging me to investigate interesting results and Herbie for keeping the final goal and deadlines in mind.

I would also like to thank my fellow graduate students in the department formerly known as AMS. Studying as a group made the coursework and qualifying exams manageable and tolerable. In addition, informal conversations and reality checks about research throughout the process were invaluable.

Finally, I would like to thank my family. First I would like to thank my wife, Sarah, for her willingness to put life on hold while I explored a significant career change, her support, and encouragement along the way. My parents were invaluable as they came out and helped with the kids numerous times throughout the process to give me more time to focus on research. I appreciate my daughter Cora for her sweet disposition and providing perspective. Lastly, I would like to thank my son Henry for being a good sleeper and generally more "Santa Cruz."

Chapter 1

Introduction

The focus of this work was the statistical and mathematical modeling of the biochemical states of DNA replication based on data observed in the context of nanopore experiments. The work here is meant to provide a more accurate and robust alternative to quick inference methods based on dwell time samples extracted using an ad hoc method [31] [32]. The motivating experiments are introduced in Section 1.1. Following Section 1.1, Section 1.2 contains a more detailed description of the models developed and organization of this thesis.

1.1 Biological Background

For somatic cell replication, the DNA of the cell must be copied. In humans, this occurs in each of their 10^{14} cells about 10^{16} times [33]. Despite the frequency and importance of replication, there are many facets of replication which are not well understood. The focus of this study was the kinetics of the translocation step and nucleotide addition cycle. Lieberman and her colleagues initially modeled these aspects of replication [32]. This study developed an alternate way to model this process to supplement the work by Lieberman et al. However, before the kinetics of the translocation step and nucleotide addition cycle could be discussed, a basic understanding of DNA construction and replication is necessary.

DNA is a linear polymer structure made up of nucleotides. The nucleotides form two complementary strands that twist around a central axis in a double-helix. Since the twisting of the nucleotide strands is unimportant for understanding the model developed, the DNA was described as if it had been untwisted and the two strands of nucleotides form a ladder-like structure. Following this analogy, each nucleotide strand is represented by "one rail of the ladder" and a "half of each rung". The "rails of the ladder" are called the backbone, and they create the support for each nucleotide strand. "Half of each rung" is one of four base compounds and these bases are the method for encoding information on the DNA. Each of the four bases has a complementary pair, and the second nucleotide strand connects the first at each rung of the ladder with the complementary base.

During transcription, the complementary nucleotide strands are separated and can be visualized by cutting the ladder in half through the middle of each "rung". Assuming fidelity, two pairs of identical DNA are created by adding the nucleotides (also referred to as deoxynucleoside triphosphate or dNTP) containing complementary base pairs to each rung of the aforementioned halved ladders.

There are a number of proteins that assist in the transcription process. This study focused on the DNA polymerase (DNAP) which is a catalyst for DNA synthesis. During replication, the DNAP is bound with a DNA strand. The DNAP acts as a catalyst to bind the dNTP to the "halved" strand.

To observe transcription, a primer was bound to the end of the DNA. Without this, there is nothing for the DNAP to bind to. Then, a ϕ29 DNAP (which was chosen because of its ability to catalyze consecutive replication pairs without additional proteins [11] [50]) was then added to the primer. A single nucleotide addition cycle is illustrated in Figure 1.1, where the DNA is depicted as a simple lattice.

The first state is labeled as state S_1 and is furthest left in Figure 1.1. The joined ends with a P on either side at the top of the structure is the primer. The center of the latter depicts the type of base with either an A (adenine), G (guanine), T (thymine), or C (cytosine). Although the entire DNAP is not depicted, the active site of the DNAP is at the -1 position and is signified with parenthesis. This state will be referred to the binary structure pre-translocation state since the structure includes the DNAP and the single DNA strand. The second state (post-translocation) is depicted in Figure 1.1 as S_{2A} where the DNA has shifted up, and the active site of the DNAP is now at the 0 position to allow the binding of dNTP. After translocation, the complementary dNTP is affixed to the polymer structure as labeled as S_{2B}. Since a third protein has been added, this state will be referred to as the post-translocation ternary structure. Having completed the nucleotide addition at position 0, the pre-translocation state (S_1) for the nucleotide addition at position 1 is labeled as NC.

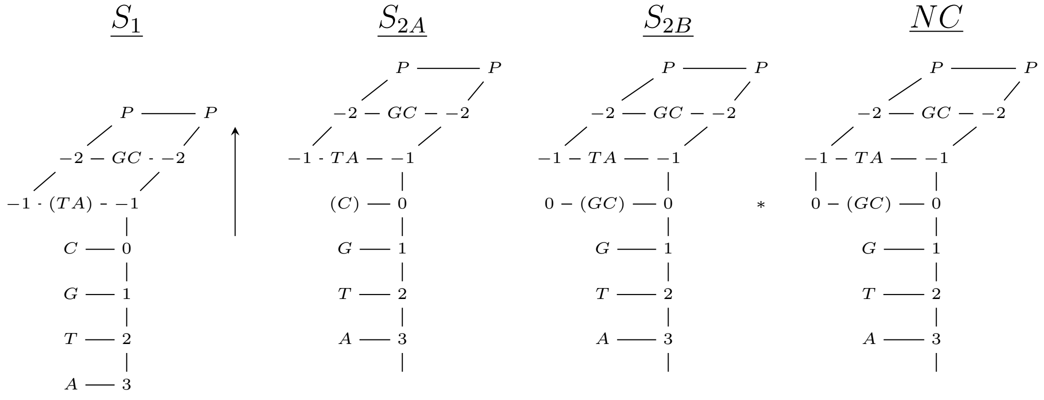


Figure 1.1: A diagram depicting a nucleotide addition cycle where the location of the active site of the DNAP is denoted by (∗)

The aforementioned states and the transition rates between them were the focus of this study. Furthermore, some of the states can be isolated. States S_1 and S_{2A} can be isolated by not adding dNTP to the solution. In this case, nothing can be bound in state S_{2A} so the system vacillates between S_1 and S_{2A}. In the experiment by Lieberman and her colleagues, evidence supported that dwell times in S_1 and S_{2A} were exponentially distributed [31]. This results in the two state space model pictured in Figure 1.2A, where r_{12} and r_{21} are the transition rate parameters from S_1 to S_{2A} and S_{2A} to S_1 respectively. Finally, S_1, S_{2A}, S_{2B} could be isolated by engineering the DNA so the dNTP could not form a covalent bond with the backbone of the DNA. In this case, the active site cannot move forward and the next cycle is not started. Similar to the previous case, the system vacillates between S_1, S_{2A}, and S_{2B}. When the dNTP was added to the system, it was seen that it is not possible for the dNTP to bind before the binary system transitioned to state S_{2A} and the dNTP did not impact r_{12} [32] [11]. In addition, the dwell time of S_{2B} is exponential. Therefore, the transitions S_{2A} to S_{2B} and S_{2B} to S_{2A} have constant transition rate parameters $k_{on}[dNTP]$ and k_{off} respectively where k_{on} is the binding affinity of the dNTP, $[dNTP]$ is the concentration of dNTP, and k_{off} is the disassociation rate of the dNTP [32]. This model is the three state model depicted in Figure 1.2B.

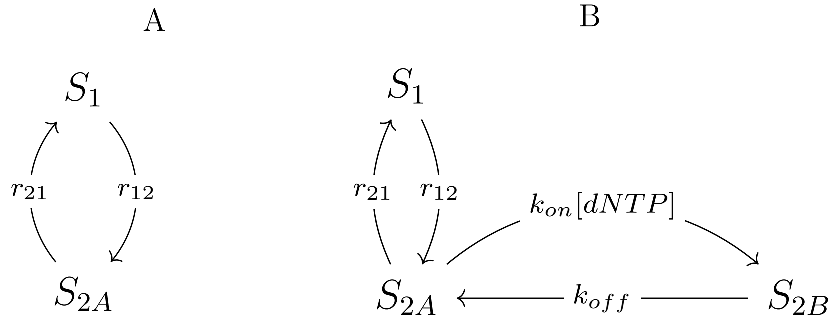


Figure 1.2: Diagrams of the two state model (A) and three state model (B)

Unfortunately, at the base pair level, it was not possible to directly track the state of the system at the detail level discussed previously. The movement of the DNAP was tracked indirectly using the $\alpha-$hemolysin nanopore [11]. It has been shown that the nanopore can be inserted into a membrane, an ionic current can be applied across the membrane, and that signal can be tracked [30] [2] as in Figure 1.3A. Then, the binary complex was placed on top of a nanopore that acted as a channel across the membrane. The DNAP is too big to pass through the channel created by the nanopore. Therefore, the DNAP sits on top of the nanopore and the single DNA strand dangles through the lumen of the nanopore [5] [30]. Finally, since abasic sites are small relative to the nucleotides, abasic sites were inserted into the DNA strand near the lumen of the nanopore. Then as the position of the abasic sites changed relative to the lumen, the relative position could be differentiated on a base pair level. Given the experimental design used by Dahl and his colleague and repeated by Liebermen, abasic sites were inserted into the DNA strand. In the experiments necessary for this work, abasic sites were inserted at positions +8 to +12 using the scaling system in Figure 1.1. With this design, an upper amplitude of ~ 32 picoamps (pA) is associated with the pre-translocation state (S_1) and a lower amplitude of ~ 26 pA is associated with the post-translocation states (S_{2A}) and (S_{2B}) [11] [31] [32] as illustrated in Figure 1.3B. Unfortunately, no difference in current could be discerned for S_{2A} and S_{2B}, but in previous studies, conclusions were made analyzing the two state data with the three state data [32].

Given the above, the two experiments produced the emissions in Figure 1.4. Utilizing data from both experiments, Lieberman and her colleagues used an ad-hoc maximum likelihood approach based on the dwell times to approximate r_{12} and r_{21} [31] and k_{on} and k_{off} [32]. However, since the dwell times in each state are exponential, time traces of the currents can be described as Markov Chains.

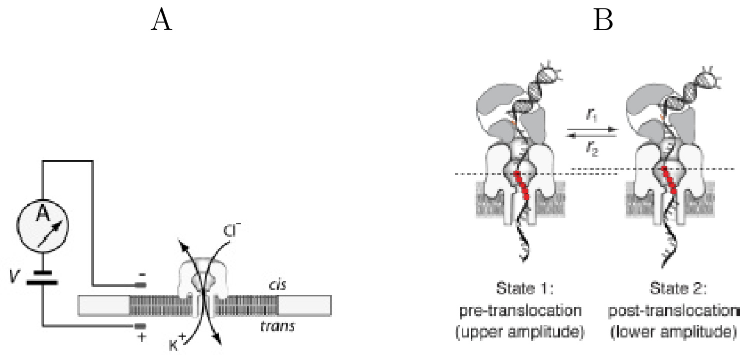


Figure 1.3: (A) The nanopore inserted into a membrane with an applied current and (B) The pre and post-translocation states when the DNAP is fixed on top of the nanopore

Using this alternate representation, a model supplemental to Lieberman and her colleagues' work was developed for this research.

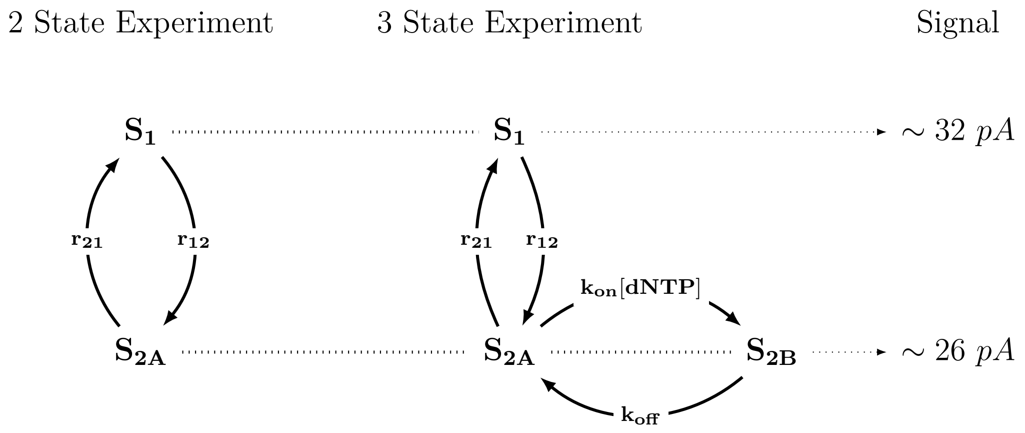


Figure 1.4: The two state and three state experiments where transitions were represented with solid arrows and emissions were represented with dotted arrows.

1.2 Description of thesis

This research focuses on the experiments described in Section 1.1. Specifically, this thesis provided additional methods for modeling these experiments and making inference on the transition rates between biochemical states.

The first experiment introduced in [31] and discussed in Section 1.1 isolated two biochemical states of DNA replication. The biochemical states and subsequent transition rates could be described in hidden Markov models. Previously developed hidden Markov modeling techniques are introduced in Chapter 2. Within that chapter, necessary preliminaries to hidden Markov modeling are introduced in Section 2.1 and Section 2.2. Although Dynamic Linear modeling was not applied to the experiments discussed, it was used to introduce hidden Markov modeling in Section 2.3 and there is an extensive discussion on data collection. The models and algorithms applied to the aforementioned two state system were introduced in Section 2.4. The methods developed prior to this research from Section 2.4 could be applied directly to the experiment discussed in [31]. The performance of these methodologies was evaluated using idealized computer generated datasets in Chapter 3.

The second experiment introduced in [32] and discussed in Section 1.1 isolated three biochemical states of DNA replication. This experiment could also be described as a hidden Markov model. However, two of the states produced indistinguishable electric signals. Following this, there was insufficient information to apply the traditional hidden Markov models introduced in Section 2.4 and this was empirically confirmed using computer generated datasets in Section 4.1. Therefore a composite state, introduced in Section 4.1.2, was considered. Using this composite state, an algorithm to compute point estimates and an MCMC sampler that simulated Bayesian posteriors of the system parameters for the experiment were developed and introduced in sections 4.2.2 and 4.2.7 respectively. Again, the methods were tested using computer generated simulations of the experiments. In Section 4.2.2, it is shown that the point estimates were biased and that bias was a noticeable contributor to the error. Therefore, a Bayesian meta-model was built to reduce the bias. The bias reduction included the inverse problem of a Gaussian process model which is introduced in Section 4.2.4 and was tailored to fit this specific problem in Section 4.2.5. The performance of the meta-model as a method for bias reduction is discussed in Section 4.2.6. Furthermore, since the meta-model was Bayesian, the credible intervals of the meta-model and the MCMC sampler could be easily compared. In Section 4.3 it is observed that both methods had very similar credible intervals and thus provided further confirmation. Lastly, the research shows that with enough data and a reasonable amount of compute time, the ranges of the credible intervals relative to the true generating parameters can be reduced.

Chapter 2

Mathematical Background

Building models of the experiments discussed in Section 1.1 required a number of tools. This chapter will discuss the methods employed that were developed prior to this research. The first and second sections of this chapter will introduce the EM algorithm and Bayesian modeling. Although these two concepts have much broader applications in the research, a mixture of geometric distributions was used as an example to introduce those subjects. The following sections will introduce hidden Markov models as they were the motivator of the algorithms developed. First, the Ornstein-Uhlenbeck process will be discussed to introduce hidden Markov models. Finally, hidden Markov models with discrete states will be introduced as the motivating biological experiments were modeled as such.

2.1 A Brief Introduction to the EM Algorithm

2.1.1 Motivation

To motivate the need for the EM method, consider the geometric distribution. Although the geometric distribution is not the most common way to introduce the

EM algorithm, it was used here because it is the most straightforward application of the EM method used directly for this project. The geometric distribution is often viewed as the number of Bernoulli trials needed to experience the first success. In this case, the probability density function was written as (2.1), where ϕ was the probability of success and d was the number of trials needed to experience the first success.

$$p(d|\phi) = (1-\phi)^{(d-1)}\phi \tag{2.1}$$

$$\hat{\phi} = \frac{H}{\sum_{h=1}^{H} d_h} \tag{2.2}$$

The maximum likelihood estimator (MLE) for ϕ given $\boldsymbol{d} = (d_1, d_2, ..., d_H)$ is well known and was listed as (2.2). Now consider the more complicated case, for each individual observation it was known that the probability of success was either φ_1 or φ_2. Then the probability density function (pdf) could be represented as (2.3) where v is an indicator variable such that $v = 1$ if the probability of success was φ_1 and $v = 0$ if probability of success was φ_2.

$$p(d|v, \varphi_1, \varphi_2) = \left[(1-\varphi_1)^{(d-1)}\varphi_1\right]^{v}\left[(1-\varphi_2)^{(d-1)}\varphi_2\right]^{(1-v)} \tag{2.3}$$

Unfortunately, it was not known whether $v = 1$ or $v = 0$ and there was not enough information to investigate further using the form described in (2.3). Therefore, a new variable w was introduced. For each trial, the probability that φ_1 was the probability of success was w and the probability that φ_2 was the probability of success was $1 - w$. Then the joint probability of d and v given φ_1, φ_2, and w could be written as (2.4).

$$p(d, v|\varphi_1, \varphi, w) = \left[w(1-\varphi_1)^{(d-1)}\varphi_1\right]^v \left[(1-w)(1-\varphi_2)^{(d-1)}\varphi_2\right]^{(1-v)} \tag{2.4}$$

Since v was unknown, the marginal distribution of d given φ_1, φ_2, and w was explored. The marginal pdf, often referred to as a mixture model, was listed as (2.6) and was derived from (2.5). In this case, the MLE was not straight forward to calculate from the log-likelihood function ($\ln \mathcal{L}$) listed as (2.7). To illustrate this, (2.8) has the first partial derivative of the log-likelihood function with respect to w. There, it can be seen that the expression contains φ_1, φ_2, and cannot easily be solved for w. Furthermore, the derivatives with respect to φ_1 and φ_2 were not more instructive.

$$p(d|\varphi_1, \varphi_2, w) = \sum_{l=0}^{1} p(d, v = l|\varphi_1, \varphi_2, w) \tag{2.5}$$

$$p(d|\varphi_1, \varphi_2, w) = w(1-\varphi_1)^{(d-1)}\varphi_1 + (1-w)(1-\varphi_2)^{(d-1)}\varphi_2 \tag{2.6}$$

$$\ln\{\mathcal{L}(\varphi_1, \varphi_2, w)\} = \sum_{h=1}^{H} \ln\left\{w(1-\varphi_1)^{(d_h-1)}\varphi_1 + (1-w)(1-\varphi_2)^{(d_h-1)}\varphi_2\right\} \tag{2.7}$$

$$\frac{\partial \ln\{\mathcal{L}(\varphi_1, \varphi_2, w)\}}{\partial w} = \sum_{h=1}^{H} \frac{(1-\varphi_1)^{(d_h-1)}\varphi_1 - (1-\varphi_2)^{(d_h-1)}\varphi_2}{w(1-\varphi_1)^{(d_h-1)}\varphi_1 + (1-w)(1-\varphi_2)^{(d_h-1)}\varphi_2} \tag{2.8}$$

Therefore, the original pdf from (2.4) was considered which results in the log-likelihood listed as (2.9). Although this allows the separation of the variables and was an improvement from the above, the MLE for $\{\varphi_1, \varphi_2, w\}$ requires knowledge of the unobserved variable $\boldsymbol{v} = (v_1, v_2, ..., v_H)$. Therefore, the EM algorithm was applied to provide knowledge of $\boldsymbol{v}$.

$$\ln\{\mathcal{L}(\varphi_1, \varphi_2, w)\} = \sum_{h=1}^{H} \ln\left\{\left[w(1-\varphi_1)^{(d_h-1)}\varphi_1\right]^{v_h} \left[(1-w)(1-\varphi_2)^{(d_h-1)}\varphi_2\right]^{(1-v_h)}\right\} \tag{2.9}$$

2.1.2 Expectation Maximization (EM) Algorithm

Although special cases of the EM algorithm had been introduced previously, the EM algorithm for general probability models was introduced by Dempster, Laird, and Rubin in 1977. The paper originally proposed the algorithm for a broad range of cases of missing data including mixture models [1] and the indicator variable in (2.9). For the general case; let $\boldsymbol{y}$ be the observed data, $\boldsymbol{v}$ be the unobserved or latent variable, and $\boldsymbol{\theta}$ be the parameters of the distribution. The EM method is an iterative algorithm where the inference on the parameters is improved in each iteration. Therefore, the inference on any variable or parameter (c) after the i^{th} iteration was denoted as $\hat{c}_i$ or $\left(\hat{c}_j\right)_i$ if the parameter contains a subscript. For each iteration $\hat{\boldsymbol{\theta}}_i$ was computed by the expression listed as (2.10) where $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{v})$ denotes the likelihood of the joint probability of $\boldsymbol{y}$ and $\boldsymbol{v}$ given $\boldsymbol{\theta}$ and $\mathcal{L}(\boldsymbol{\theta})$ denotes the likelihood of the marginal probability of $\boldsymbol{y}$ given $\boldsymbol{\theta}$. Since the EM algorithm is an iterative algorithm, an initial value must be chosen $(\hat{\boldsymbol{\theta}}_0)$ with reasonable values. Then, $\hat{\boldsymbol{\theta}}_i$ was computed using (2.10) where $\hat{\boldsymbol{\theta}}_{i-1}$ was the initial values if $i = 1$ or the values from the previous iteration if $i > 1$. Given $\hat{\boldsymbol{\theta}}_0$ was reasonable, the iterations of (2.10) will converge to $\hat{\boldsymbol{\theta}}$. For a basic justification of the algorthm, [1], [17], and [37] prove that $\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}$ is monotonic. Unfortunately, that makes the EM algorithm a local optimizer and a poor choice for $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$ can result in a solution that is not a global maximum. For further reading, [37] contains extended conversation on analytical properties and convergence rates of the EM algorithm past what was written in [1]. [56] contains more information on mixture models and the specific application of the EM algorithm on finite mixture models.

$$\hat{\boldsymbol{\theta}}_i = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \, \mathbb{E}_{\left(\boldsymbol{v}|\boldsymbol{y},\hat{\boldsymbol{\theta}}_{i-1}\right)}[\ln \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{v})] \tag{2.10}$$

The computation of (2.10) was split up into two steps, an expectation step and a maximization step. To formulate these steps, consider the joint log-likelihood. The joint log-likelihood can be written as the sum of terms with the form $f_j(\boldsymbol{v})g_j(\boldsymbol{\theta}, \boldsymbol{y})$. Using this form, the expectation step consisted of computing $\mathbb{E}_{\left(\boldsymbol{v}|\hat{\boldsymbol{\theta}}_{i-1},\boldsymbol{y}\right)}[f_j(\boldsymbol{v})]$ for each $f_j(\boldsymbol{v})$ in the joint log likelihood. After the expectation is completed, the maximization step computed $\hat{\boldsymbol{\theta}}_i$ by maximizing $\ln \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{v})$ given $f_j(\boldsymbol{v}) = \mathbb{E}_{\left(\boldsymbol{v}|\hat{\boldsymbol{\theta}}_{i-1},\boldsymbol{y}\right)}[f_j(\boldsymbol{v})]$ for each $f_j(\boldsymbol{v})$ in the joint log-likelihood. $\hat{\boldsymbol{\theta}}_i$ was computed iteratively until $\hat{\boldsymbol{\theta}}_i$ had converged to a value. Computationally this was approximated by using the stopping condition $||\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1}|| \leq \delta$ where $||\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1}||$ denotes a distance metric and δ was an arbitrary small number. In this research, the infinity norm was used which was denoted $||\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1}||_\infty$. The general algorithm was listed as (2.11).

EM Algorithm (2.11)

```
I   Pick $\hat{\boldsymbol{\theta}}_0$

II  Set $i = 0$ and define $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$.

III While $\left|\left|\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1}\right|\right|_\infty > \delta$

    (a) $i = i + 1$

    (b) Compute $\mathbb{E}_{\left(\boldsymbol{v}|\hat{\boldsymbol{\theta}}_{i-1},\boldsymbol{y}\right)}[f_j(\boldsymbol{v})]$ for each $f_j(\boldsymbol{v})$ in the joint log-likelihood

    (c) Compute $\hat{\boldsymbol{\theta}}_i$ that maximizes $\ln \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{v})$ given $f_j(\boldsymbol{v}) = \mathbb{E}_{\left(\boldsymbol{v}|\hat{\boldsymbol{\theta}}_{i-1},\boldsymbol{y}\right)}[f_j(\boldsymbol{v})]$
        for each $f_j(\boldsymbol{v})$ in the joint log-likelihood.
```

To better illustrate (2.11), consider the mixture of two geometric distributions. The joint log-likelihood was rearranged in (2.12).

$$
\begin{aligned}
\ln\left\{\mathcal{L}(\varphi_1,\varphi_2,\mathbf{w})\right\} &= \sum_{h=1}^{H} \ln\left\{\left[w(1-\varphi_1)^{(d_h-1)}\varphi_1\right]^{v_h}\left[(1-w)(1-\varphi_2)^{(d_h-1)}\varphi_2\right]^{(1-v_h)}\right\} \\
&= \sum_{h=1}^{H}\left[v_h\left(\ln(w)+(1-d_h)\ln(1-\varphi_1)+\ln(\varphi_1)\right)+...\right. \qquad (2.12) \\
&\quad \left. ...+(1-v_h)\left(\ln(1-w)+(1-d_h)\ln(1-\varphi_2)+\ln(\varphi_2)\right)\right]
\end{aligned}
$$

From (2.12), it can be seen the $f_j(\boldsymbol{v})$ were simply v_h. Therefore, for a mixture of two geometric distributions, the Expectation step for the i^{th} iteration was computing $\left(\mathbb{E}_{\left(\boldsymbol{v}|\hat{\boldsymbol{\theta}}_{i-1},\boldsymbol{y}\right)}[v_h]\right)_i$ denoted as $(\mathbb{E}[v_h])_i$ for $h=1$ to H where $(\mathbb{E}[v_h])_i$ was computed in (2.15) to (2.16). For brevity, let $p\left(d_h|v_h=1,\hat{\boldsymbol{\theta}}_{i-1}\right)$ and $p\left(d_h|v_h=0\right.$, $\left.\hat{\boldsymbol{\theta}}_{i-1}\right)$ be defined as (2.13) and (2.14).

$$p\left(d_h|v_h=1,\hat{\boldsymbol{\theta}}_{i-1}\right) = \left(1-\left\{\hat{\varphi}_1\right\}_{i-1}\right)^{(d_h-1)}\left\{\hat{\varphi}_1\right\}_{i-1} \qquad (2.13)$$

$$p\left(d_h|v_h=0,\hat{\boldsymbol{\theta}}_{i-1}\right) = \left(1-\left\{\hat{\varphi}_2\right\}_{i-1}\right)^{(d_h-1)}\left\{\hat{\varphi}_2\right\}_{i-1} \qquad (2.14)$$

$$
\begin{aligned}
(\mathbb{E}[v_h])_i &= \sum_{v_h=0}^{1} v_h p\left(v_h|d_h,\hat{\boldsymbol{\theta}}_{i-1}\right) \qquad (2.15) \\
(\mathbb{E}[v_h])_i &= \sum_{v_h=0}^{1} v_h \frac{p\left(d_h,v_h|\hat{\boldsymbol{\theta}}_{i-1}\right)}{p\left(d_h|\hat{\boldsymbol{\theta}}_{i-1}\right)} \\
(\mathbb{E}[v_h])_i &= \sum_{v_h=0}^{1}\left(v_h\frac{\left[p\left(d_h|v_h=1,\hat{\boldsymbol{\theta}}_{i-1}\right)\right]^{v_h}\left[p\left(d_h,v_h=0|\hat{\boldsymbol{\theta}}_{i-1}\right)\right]^{(1-v_h)}}{p\left(d_h,v_h=1|\hat{\boldsymbol{\theta}}_{i-1}\right)+p\left(d_h|v_h=0,\hat{\boldsymbol{\theta}}_{i-1}\right)}\right) \\
(\mathbb{E}[v_h])_i &= \frac{\hat{w}_{i-1}\left[p\left(d_h|v_h=1,\hat{\boldsymbol{\theta}}_{i-1}\right)\right]}{\hat{w}_{i-1}\left[p\left(d_h|v_h=1,\hat{\boldsymbol{\theta}}_{i-1}\right)\right]+(1-\hat{w}_{i-1})\left[p\left(d_h|v_h=0,\hat{\boldsymbol{\theta}}_{i-1}\right)\right]} \qquad (2.16)
\end{aligned}
$$

Using (2.16), (2.10) was computed by maximizing (2.17). Furthermore, the maximization for (2.17) was found by computing the MLE for $(\hat{\varphi}_1)_i$, $(\hat{\varphi}_2)_i$, and $\hat{w}_i$ which were listed as (2.18), (2.19), and (2.20) respectively.

$$\begin{aligned}
\{\mathbb{E}\left[\ln \mathcal{L}\left(\boldsymbol{\theta}\right)\right]\}_i &= \ln \mathcal{L}\left(\boldsymbol{\theta}, \{\mathbb{E}[v_h]\}_i\right) \\
\{\mathbb{E}\left[\ln \mathcal{L}\left(\boldsymbol{\theta}\right)\right]\}_i &= \sum_{h=1}^{H} \left[\{\mathbb{E}[v_h]\}_i \left(\ln(w) + (1-d_h)\ln(1-\varphi_1) + \ln(\varphi_1)\right) + ... \right. \qquad (2.17) \\
&\left. ...\left(1 - \{\mathbb{E}[v_h]\}_i\right)\left(\ln(1-w) + (1-d_h)\ln(1-\varphi_2) + \ln(\varphi_2)\right)\right]
\end{aligned}$$

$$(\hat{\varphi}_1)_i = \frac{\sum_{h=1}^{n} \{\mathbb{E}[v_h]\}_i}{\sum_{h=1}^{n} \{\mathbb{E}[v_h]\}_i \,(d_h)} \qquad (2.18)$$

$$(\hat{\varphi}_2)_i = \frac{\sum_{h=1}^{n} 1 - \{\mathbb{E}[v_h]\}_i}{\sum_{h=1}^{n} \left(1 - \{\mathbb{E}[v_h]\}_i\right)(d_h)} \qquad (2.19)$$

$$\hat{w}_i = \frac{\sum_{h=1}^{n} \{\mathbb{E}[v_h]\}_i}{n} \qquad (2.20)$$

Finally, the EM algorithm for a mixture of two geometric distribution was listed below.

EM Algorithm for a Mixture of Two Geometric Distributions (2.21)

```
I Pick $\hat{\boldsymbol{\theta}}_0$
II Set $i = 0$ and define $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$.
III While $\left|\left|\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1}\right|\right|_\infty > \delta$
    (a) $i = i + 1$
    (b) for $h = 1 : H$
        i. Calculate $(\mathbb{E}[v_h])_i$ using (2.16).
    (c) Calculate $(\hat{\varphi}_1)_i$ using (2.18).
    (d) Calculate $(\hat{\varphi}_2)_i$ using (2.19).
    (e) Calculate $\hat{w}_i$ using (2.20).
```

2.2 A Brief Introduction to Bayesian Modeling

In the last section, inference was made on $\hat{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ is a parameter vector with single values. However, one could also view $\boldsymbol{\theta}$ as a vector of random variables. Given this structure, the inference provides much more detailed information about the range or distribution of $\boldsymbol{\theta}$. Unfortunately, this extra information has a cost. Bayesian inference is much more computationally expensive than the inference on $\hat{\boldsymbol{\theta}}$ as discussed in the last section. The following section gives a brief introduction to Bayesian Modeling but is far from comprehensive. For more reading, [17] is a good book on the subject.

2.2.1 Bayesian Inference

Bayes' Theorem (2.22)

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}$$

This research focuses on two types of inference, inference on the parameters ($\boldsymbol{\theta}$) given the data ($\boldsymbol{y}$) and inference on future data ($\boldsymbol{y}_*$) given the data. The two types of inference were denoted as $\boldsymbol{\theta}|\boldsymbol{y}$ and $\boldsymbol{y}_*|\boldsymbol{y}$ respectively. To illustrate this, consider the geometric distribution listed again as (2.26). The posterior of $\boldsymbol{\theta}$ would include inference made on w, φ_1, and φ_2. The predictive distribution, in this case, would include inference on $\boldsymbol{d}_*$ where $\boldsymbol{d}_*$ is predicting the distribution of a future dataset. Inference on both $\boldsymbol{\theta}$ and $\boldsymbol{y}_*$ (or $\boldsymbol{d}_*$ in the previous example) are dependent on Bayes' Theorem, the namesake of the model. Inference on $p(\boldsymbol{\theta}|\boldsymbol{y})$ is referred to as the posterior density. To derive the posterior density in this model, a prior distribution ($p(\boldsymbol{\theta})$) was required. The prior distribution signifies the distribution of $\boldsymbol{\theta}$ before the existence of the dataset. When one picks a prior, often two separate factors are considered. The first factor is to choose a prior that results in a posterior density which is not too computationally intensive.

The second factor is the information provided by the prior. In many cases, little is known about $\boldsymbol{\theta}$, and the prior should reflect that by contributing little or nothing to the posterior distribution. However, there are cases where information about $\boldsymbol{\theta}$ is known before data collection. In that case, the prior should reflect that information. Once the prior is chosen, the general case of the posterior density is proportional to (2.23) which follows directly from (2.22).

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})} \\ p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \end{aligned} \tag{2.23}$$

Given the posterior distribution from (2.23), the predictive distribution follows immediately. The predictive distribution is (2.24) where $\boldsymbol{\vartheta}$ is the parameter space for $\boldsymbol{\theta}$ assuming $\boldsymbol{\vartheta}$ a continuous subspace.

$$p(\boldsymbol{y}_*|\boldsymbol{\theta}) = \int_{\boldsymbol{\theta}\in\boldsymbol{\vartheta}} p(\boldsymbol{y}_*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\vartheta} \tag{2.24}$$

There are a few cases where (2.23) and (2.24) can be computed analytically. One such case is when a conjugate prior for $\boldsymbol{\theta}$ is used. A conjugate prior is a prior such that the posterior of $\boldsymbol{\theta}$ results in the same type of distribution as the prior. However, this method is not applicable for most cases including the inference made for this research. In these cases, numerical methods such as MCMC are applied.

2.2.2 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a common way to numerically estimate the distributions from (2.23) and (2.24) when $p(\boldsymbol{\theta}|\boldsymbol{y})$ is intractable. MCMC simulates N random draws from the distribution of the posterior (or predictive) where the i^{th} draw from the posterior of parameter c will be denoted as c_i or $(c_j)_i$ if c has a subscript. This differs from $\hat{\boldsymbol{\theta}}_i$ because each $\hat{\boldsymbol{\theta}}_i$ improves the estimate of the true value of $\boldsymbol{\theta}$. As stated earlier, in Bayesian inference $\boldsymbol{\theta}$ is a random variable and has a distribution. Following this, each MCMC draw $(\boldsymbol{\theta}_i)$ provides one more simulation from the distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$. Therefore, $\boldsymbol{\theta}_i$ is not expected to be closer to the mean of $\boldsymbol{\theta}$ than $\boldsymbol{\theta}_{i-1}$. Each simulation of $\boldsymbol{\theta}_i$ combined with the previous simulations provides a more complete picture of the distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$.

This section will have some practical information on implementing two different types of MCMC samplers, but [47] describes the development of MCMC throughout history. [17] has a basic introduction on MCMC which includes more on convergence, sufficient sample sizes, efficient computation, and alternatives. [3] includes more justification of MCMC methods in a short format. [46] includes all of the aforementioned subjects in more detail.

Gibbs Sampler

The Gibbs sampler is a special case of the Metropolis-Hasting algorithm which is described next. Since the Gibbs sampler is a special case of the Metropolis-Hastings algorithm, it is often introduced after the Metropolis-Hastings algorithm. However, since the theory was not discussed here, Gibbs was introduced first as it is much easier to implement. There are many papers that included special cases and necessary building blocks but [18] is considered to be the seminal paper on Gibbs sampling. Furthermore, for more information, Gibbs sampling is discussed in all the resources provided in the previous subsection.

For the general implementation of the Gibbs sampler, let $\boldsymbol{y}$ be the observed data and $\boldsymbol{\theta} = (\tau_1, \tau_2, ..., \tau_M)$ be the parameters of the model. For a Gibbs sampler, each τ_m is sampled separately. Then, the i^{th} random draw for τ_m is made from $p\left(\tau_m|\boldsymbol{y}, (\boldsymbol{\theta}_{-m}))_{i-1}\right)$ where $(\boldsymbol{\theta}_{-m})_{i-1} = \left((\tau_1)_{i-1}, (\tau_2)_{i-1}, ..., (\tau_{m-1})_{i-1}, (\tau_{m+1})_{i-1}, ..., (\tau_M)_{i-1}\right)$ or all $\boldsymbol{\theta}_{i-1}$ except τ_m. For this, each conditional posterior distribution $\left(p\left(\tau_m|\boldsymbol{y}, (\boldsymbol{\theta}_{-m})_{i-1}\right)\right)$, must be able to be sampled from. Given that $p\left(\tau_m|\boldsymbol{y},(\boldsymbol{\theta}_{-m})_{i-1}\right)$ can be sampled from, the draw for τ_m can be generated using a random number generator. A general algorithm for a Gibbs sampler which draws N samples from the posterior was listed as (2.25).

A general Gibbs sampler (2.25)

```
I Make an initial guess or choose values for $\boldsymbol{\theta}_0$
II for i=1:N
    (a) for m=1:M
        i. Draw $(\tau_m)_i \sim p\left(\tau_m|\boldsymbol{y},(\boldsymbol{\theta}_{-m})_{i-1}\right)$
    (b) end
III end
```

To illustrate this, consider the joint probability of a mixture of two geometric distributions introduced in Section 2.1 and listed as (2.26).

$$p(\boldsymbol{d},\boldsymbol{v}|w,\varphi_1,\varphi_2)=\prod_{h=1}^{H}\left[w(1-\varphi_1)^{(d_h-1)}\varphi_1\right]^{v_h}\left[(1-w)(1-\varphi_2)^{(d_h-1)}\varphi_2\right]^{1-v_h} \tag{2.26}$$

For more explanation on the distribution see Section 2.1. As in Section 2.1, only $\boldsymbol{d}$ was known and $\boldsymbol{v}$ was latent. Therefore, in addition to drawing from the conditional posterior of φ_1, φ_2, and w, draws from the conditional posterior of $\boldsymbol{v}$ must also be made. The Gibbs sampler was listed as (2.27), and the distributions referenced in the sampler were derived and listed in the following text.

A Gibbs sampler for a mixture of geometric distributions (2.27)

```
I Make an initial guess or choose values for $(\varphi_1)_0$, $(\varphi_2)_0$, and $w_0$.
II for $h = 1 : H$
```

```
        (a) draw $(v_h)_0$ from (2.38)
  III end
  IV for $i = 1 : N$
        (a) draw $(\varphi_1)_i$ from (2.32)
        (b) draw $(\varphi_2)_i$ from (2.33)
        (c) draw $w_i$ from (2.31)
        (d) for $h = 1 : H$
              i. draw $(v_h)_i$ from (2.38)
        (e) end
  V end
```

As stated, the prior distributions must be chosen so that each conditional posterior distribution can be sampled. Therefore, each prior distribution satisfied this requirement. First, the posterior of w was addressed by applying (2.23) which resulted in (2.28). Unfortunately, picking a prior of the form $p(w|\boldsymbol{d}, \boldsymbol{v}, \varphi_1, \varphi_2)$ was difficult. To resolve this, a prior for w that was independent of φ_1, φ_2, and $\boldsymbol{v}$ was used. Then $p(w|\varphi_1, \varphi_2) = p(w)$ resulting in (2.29).

$$p(w|\boldsymbol{d}, \boldsymbol{v}, \varphi_1, \varphi_2) \propto p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w)p(w|\varphi_1, \varphi_2) \tag{2.28}$$

$$p(w|\boldsymbol{d}, \boldsymbol{v}, \varphi_1, \varphi_2) \propto p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w)p(w) \tag{2.29}$$

To ensure a posterior that could be sampled, a conditionally conjugate prior was chosen. Much like a conjugate prior, a conditionally conjugate prior is a prior such that the conditional posterior is the same type of distribution as the conditionally conjugate prior. The distribution that was conditionally conjugate to the joint probability in the conditional posterior of w was a beta distribution. In the following, a beta distribution with parameters α and β was represented by $\mathcal{B}(\alpha, \beta)$ which was defined by the probability density function in 2.30.

The Beta distribution ($\mathcal{B}(\alpha, \beta)$) (2.30)

$$
\begin{aligned}
f(x; \alpha, \beta) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \\
x &\in [0, 1] \\
\alpha &> 0 \\
\beta &> 0 \\
\Gamma(x) &= \int_0^\infty y^{x-1} e^{-y} dy
\end{aligned}
$$

The posterior resulting from (2.29) was listed as (2.31), and the full derivation can be found in Appendix A.1.1.

$$
\begin{aligned}
p(w|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w}) &\propto p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w)p(w) \\
w &\sim \mathcal{B}(\alpha_w, \beta_w) \\
w|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w} &\sim \mathcal{B}_w\left(\sum_{h=1}^{H}(v_h) + \alpha_w, \sum_{h=1}^{H}(1 - v_h) + \beta_w\right)
\end{aligned}
\tag{2.31}
$$

φ_1 and φ_2 were dealt with simultaneously because solving for the posterior of each was essentially the same problem. For φ_1 and φ_2, a conditionally conjugate prior was employed. For both, the conditionally conjugate prior was also a beta distribution. The resulting posteriors were listed as (2.32) and (2.33). Full derivations can be found in Appendix A.1.2.

$$p(\varphi_i|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-\varphi_i}) \propto p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w)p(\varphi_i)$$

$$\varphi_i \sim \mathcal{B}(\alpha_{\varphi_i}, \beta_{\varphi_i})$$

$$\varphi_1|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-\varphi_1} \sim \mathcal{B}\left(\sum_{h=1}^{H}(v_h) + \alpha_{\varphi_1}, \sum_{h=1}^{H}(v_h(d_h - 1)) + \beta_{\varphi_1}\right) \tag{2.32}$$

$$\varphi_2|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-\varphi_2} \sim \mathcal{B}\left(\sum_{h=1}^{H}(1 - v_h) + \alpha_{\varphi_2}, \sum_{h=1}^{H}((1 - v_h)(d_h - 1)) + \beta_{\varphi_2}\right) \tag{2.33}$$

The amount of information provided by the priors above can be controlled through the parameters of the priors (α and β). In this work, there was not initial information for w, φ_1, and φ_2. Therefore, by choosing $\alpha = \beta = 1$ for all priors, the priors were flat.

Finally, v_h must be drawn even though v_h was an unobserved variable or a latent variable. Unlike the posteriors for φ_1, φ_2, and w, a prior did not need to be selected for $\boldsymbol{v}$. That is because the prior for $\boldsymbol{v}$ naturally occurred in the model used. (2.34) and (2.35) shows that the joint probability was equal to the product of the conditional probability of d and the prior on v_h. Furthermore, (2.34) was equal to the right hand side of (2.23) making (2.34) proportional to the conditional posterior of v_h.

$$p(d_h, v_h|w, \varphi_1, \varphi_2) = p(d_h|v_h, w, \varphi_1, \varphi_2)p(v_h|w, \varphi_1, \varphi_2) \tag{2.34}$$

$$p(v_h|\boldsymbol{d}, \boldsymbol{\theta}) \propto p(d_h|v_h, w, \varphi_1, \varphi_2)p(v_h|w, \varphi_1, \varphi_2) \tag{2.35}$$

$$p(d_h|v_h, w, \varphi_1, \varphi_2) = \left[(1 - \varphi_1)^{(d_h - 1)}\varphi_1\right]^{v_h}\left[(1 - \varphi_2)^{(d_h - 1)}\varphi_2\right]^{1 - v_h} \tag{2.36}$$

$$p(v_h|w, \varphi_1, \varphi_2) = p(v_h|w) = w^{v_h}(1 - w)^{1 - v_h} \tag{2.37}$$

Using $p(d_h|v_h, w, \varphi_1, \varphi_2)$ as defined in Section 2.1 and listed as (2.36), the prior on v_h was *Bernoulli*(w) as written in (2.37). The posterior of v_h was listed as (2.38) and the full derivation can be found in Appendix A.1.3.

$$p(v_h|d_h, \boldsymbol{\theta}) \propto p(\boldsymbol{d}|\varphi_1, \varphi_2, w, v_h)p(v_h|w)$$

$$v_h|d_h, \boldsymbol{\theta} \sim Bernoulli\left(\frac{\left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]}{w(1-\varphi_1)^{d_h-1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h-1}\varphi_2}\right) \tag{2.38}$$

Metroplois-Hastings algorithm

In many cases, it is not possible to sample from the conditional posterior for all τ_m, regardless of the choice for priors. Given this, the Metropolis-Hasting algorithm could be employed. A special case of the algorithm was developed by Metropolis and his colleagues in 1953 [38] and was generalized by Hastings in 1970 [22]. As stated above, [17] , [3], and [46] are good resources beyond this text and the seminal papers on this subject. For the general case, the notation from Section 2.2.2 was used. Since $\boldsymbol{\theta}_i$ could not be drawn from a distribution as in the Gibbs sampler, a proposal ($\boldsymbol{\theta}_p$) was generated. This proposal was generated using a "jumping distribution" given the previous value for $\boldsymbol{\theta}$ which was denoted at $J\left(\boldsymbol{\theta}_p|\boldsymbol{\theta}_{i-1}\right)$. Then, a value r, which compared the probability of $\boldsymbol{\theta}_p$ with $\boldsymbol{\theta}_{i-1}$ was computed using the formula in (2.39). If $r \geq 1$ (ie $\boldsymbol{\theta}_p$ was more likely to occur than $\boldsymbol{\theta}_{i-1}$), than $\boldsymbol{\theta}_i = \boldsymbol{\theta}_p$. If $r \leq 1$, than $\boldsymbol{\theta}_i = \boldsymbol{\theta}_p$ with probability r, otherwise $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1}$. Like Gibbs sampling, this was continued for N draws.

The general Metropolis-Hastings algorithm. (2.39)

```
I Make an initial guess or choose values for $\boldsymbol{\theta}_0$
II for i=1:N
    (a) Generate $\boldsymbol{\theta}_p$ from $J\left(\boldsymbol{\theta}_p|\boldsymbol{\theta}_{i-1}\right)$
    (b) Set $r = \frac{p(\boldsymbol{\theta}_p|\boldsymbol{y})/J(\boldsymbol{\theta}_p|\boldsymbol{\theta}_{i-1})}{p(\boldsymbol{\theta}_{i-1}|\boldsymbol{y})/J(\boldsymbol{\theta}_{i-1}|\boldsymbol{\theta}_p)}$
    (c) Get $c$ where $c \sim unif(0,1)$
    (d) If $r \geq c$ than $\boldsymbol{\theta}_i = \boldsymbol{\theta}_p$, otherwise $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1}$
III end
```

By varying $J\left(\boldsymbol{\theta}_p|\boldsymbol{\theta}_{i-1}\right)$, it will change the acceptance probability of $\boldsymbol{\theta}_p$. The acceptance probability only affects the efficiency of mixing. Therefore, given enough draws, the algorithm should simulate the true distribution regardless of acceptance rate. However, for most one-dimensional problems, an acceptance rate of 0.44 should be targeted while a rate of 0.23 should be targeted for most cases when proposing a vector as above. More on this and assessing convergence can be found in [17]. Finally, the Metropolis-Hastings algorithm can be used to compute the value of a single conditional posterior distribution within a Gibbs sampler. This was the setting in which Metropolis-Hasting was usually applied for this research. To illustrate this, consider the mixture of two geometric distributions. The calculation for (2.31) could be replaced with a Metropolis-Hastings step. For this, a jumping distribution was chosen $(J\left(w_p|w_{i-1}\right))$. A common choice is $w_p \sim \mathcal{N}\left(w_{i-1}, \sigma^2\right)$ where $\mathcal{N}$ denotes the normal distribution. However, $0 \leq w \leq 1$, so the Normal distribution was not viable for this example. This was rectified by using the truncated normal distribution on 0 to 1 with mean w_{i-1} and variance σ^2 which was denoted $\mathcal{TN}\left(w_{i-1}, \sigma^2; (0,1)\right)$ Then, the posterior of (2.26) could be simulated using (2.27) except (2.31) was replaced by (2.40). In (2.40) $J(w_p|w_{i-1})$ represents the probability density function for $w_p \sim \mathcal{TN}\left(w_{i-1}, \sigma^2; (0,1)\right)$ and $J(w_{i-1}|w_p)$ represents the probability density function for $w_{i-1} \sim \mathcal{TN}\left(w_p, \sigma^2; (0,1)\right)$.

Metropolis-Hastings step for w_i. (2.40)

```
  I Generate $w_p \sim \mathcal{TN}\left(w_{i-1}, \sigma^2; (0,1)\right)$
 II Set $r = \frac{p(w_p|\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta}_{-w})/J(w_p|w_{i-1})}{p(w_{i-1}|\boldsymbol{y},\boldsymbol{v},\boldsymbol{\theta}_{-w})/J(w_{i-1}|w_p)}$
III Get $c$ where $c \sim unif(0,1)$
 IV If $r \geq c$ than $w_i = w_p$, otherwise $w_i = w_{i-1}$
```

With this, the acceptance rates of w_p can be changed by choosing different values for σ^2. Finally, although this change would simulate the posterior, the Gibbs sampler listed originally is more efficient, and the Metropolis-Hasting algorithm should only be applied when necessary.

Sampling from the posterior predictive distribution

The posterior predictive distribution was defined as (2.24). There are cases where posterior predictive distribution can be calculated analytically, but many cases including those from this research are simulated. Given the posterior distribution with N samples, sampling N draws from the posterior predictive distribution is not difficult. The algorithm for sampling was listed as (2.41).

Simulating N draws from the distribution of $p(\boldsymbol{y}_*|\boldsymbol{y})$. (2.41)

```
I   Simulate $N$ draws from the distribution of $p\left(\boldsymbol{\theta}|\boldsymbol{y}\right)$
II  for $i = 1 : N$
      (a) Sample $\left(\boldsymbol{y}_*\right)_i \sim p\left(\boldsymbol{y}_*|\boldsymbol{\theta}_i\right)$
III end
```

Furthermore, this sampler could be listed as an additional step in (2.25), as opposed to a separate sampler. To illustrate predictive sampling, the mixture of two geometric distributions was employed. Let $\boldsymbol{d}_* = (d_{*1}, d_{*2}, ...d_{*H})$ and $\left(d_{*h}\right)_i$ be the i^{th} draw from the posterior of d_{*h}. Then $\left(d_{*h}\right)_i$ can be sampled from $Geo\left(\left(\varphi_1\right)_i\right)$ if $\left(v_h\right)_i = 1$ or $Geo\left(\left(\varphi_2\right)_i\right)$ if $\left(v_h\right)_i = 0$ where $Geo(c)$ is a geometric distribution with a success rate of c. The algorithm for this sampler is found in (2.42).

Simulating N draws from the distribution of $p(\boldsymbol{d}_*|\boldsymbol{d})$. (2.42)

```
I   Simulate $N$ draws from the distribution of $p\left(\varphi_1, \varphi_2, w, \boldsymbol{v}|\boldsymbol{d}\right)$
II  for $i = 1 : N$
```

```
    (a) for $h = 1 : H$
        i. if $(v_h)_i = 1$ draw $(d_{*h})_i \sim Geo\left((\varphi_1)_i\right)$
        ii. if $(v_h)_i = 0$ draw $(d_{*h})_i \sim Geo\left((\varphi_2)_i\right)$
III end
```

This sampler essentially creates HN draws from the posterior where H was the size of the original dataset and N was the number of draws from the posterior. In most cases, it is not necessary to draw a vector of H predictive values for each i. However, using this format for the predictive distribution improves results for finite mixture models. For more information, [52] contains extended discussion on this topic in the context of mixture models.

The posterior predictive samples can be used as justification for the model as well as predicting future observations. For this research, it was used for prediction, but [17] contains more discussion on evaluating a model using the posterior predictive distribution. Furthermore, he discusses the predictive model in regards to many different distributions.

2.3 Parameter Inference on an Orstein-Uhlenbeck Process

$$dX = -B\left(X - \mu_{ou}\right)dt + \gamma dW \tag{2.43}$$

The Orstein-Ulenbeck process is a continuous time process that is described by the stochastic differential equation in (2.43). Although the process is a continuous time process, the noisy measurement of X occurred at discrete times. This section discusses the methodology of using those measurements to make inference on the parameters of the governing system.

The seminal research was done by Ornstein and Uhlenbeck to model Brownian motion [57]. Although the Orstein-Ulenbeck process is a common modeling tool with many applications, the model itself was not a focus area for this research. The Orstein-Ulenbeck process was used here to introduce Hidden Markov models (HMMs) and to explore how inference on the HMM affected inference on B and γ. For this, the special case where $\mu_{ou} = 0$ was considered and was listed as (2.44).

$$\begin{aligned} dX &= -BXdt + \gamma dW \\ B &\geq 0 \end{aligned} \tag{2.44}$$

Before delving into inference, it is helpful to have a basic understanding of (2.44). Following that, the Ornstein-Ulenbeck was described as a type of random walk. The first term ($-BXdt$) is not random and pulls X back toward zero at a rate proportional to the product of the distance from zero and time elapsed (dt). The second term contributes the randomness where γ is a constant and W is a Wiener process defined by the properties in (2.45).

Properties of Wiener Process (W) (2.45)

I $W(0) = 0$

II For $t_1 \leq t_2$, $W(t_2) - W(t_1) \sim N(0, t_2 - t_1)$

III For $t_1 \leq t_2 \leq t_3 \leq t_4$, $W(t_2) - W(t_1)$ and $W(t_4) - W(t_3)$ are independent.

2.3.1 Hidden Markov Models

Rather than having observations made continuously, the data includes observations of X at a discrete set of times t_1, t_2,...., and t_T where $t_i - t_{i-1} = \Delta t$ for all $i \in \{2, 3, ..., T\}$.

The discrete case of (2.44) was listed as (2.46) and (2.47) where x_i is $X(t_i)$. The mapping to k and σ^2_η are (2.48) and (2.49) and derivations can be found in the Appendices B.1.1 and B.1.2 respectively. The derivation follows the outline of the derivation from [57]. It is less concise, but the derivation used does not require knowledge specific to stochastic differential equations as that was not a focus of this work.

$$x_1 \sim \mathcal{N}_{x_1}\left(\mu_0, \sigma^2_\eta\right) \tag{2.46}$$

$$x_i = kx_{i-1} + \eta_i \qquad \eta_i \sim N(0, \sigma^2_\eta) \tag{2.47}$$

$$k = e^{-B(\Delta t)} \tag{2.48}$$

$$\sigma^2_\eta = \frac{\gamma^2}{2B}\left[1 - e^{-2B(\Delta t)}\right] \tag{2.49}$$

A Markov model is a model such that $p(x_i|x_{i-1}, x_{i-2}, ..., x_1) = p(x_i|x_{i-1})$ or x_i only depends on the previous state. From (2.47), $p(x_i|x_{i-1}, x_{i-2}, ..., x_1) = p(x_i|x_{i-1}) = \mathcal{N}(kx_{i-1}, \sigma^2_\eta)$, and therefore the discrete system is Markovian. A Hidden Markov model is where x_i is not observed directly. For (2.47), assume there was random measurement error of the true state x_i. Then, the dataset included all y_i where $y_i \sim \mathcal{N}(x_i, \sigma^2_\varsigma)$ and the true x_i is unknown. This results in (2.50) and this combined with (2.46) and (2.47) is a special case of Hidden Markov model called a dynamic linear model (DLM).

$$y_i = x_i + \varsigma_i \tag{2.50}$$

$$\varsigma \sim \mathcal{N}(0, \sigma^2_\varsigma)$$

2.3.2 Making Inference on a DLM

A dynamic linear model (DLM) (2.51)

$$
\begin{aligned}
& x_1 \sim \mathcal{N}_{x_1}\left(\mu_0, \sigma_\eta^2\right) & & \\
& x_i = kx_{i-1} + \eta_i & & \eta_i \sim N(0, \sigma_\eta^2) \\
& y_i = x_i + \varsigma_i & & \varsigma \sim \mathcal{N}(0, \sigma_\varsigma^2)
\end{aligned}
$$

Recall, the interest was making inference on B and γ, but direct inference given the data was not possible. Instead, inference was made on the parameters of the DLM which included $\boldsymbol{\theta} = (k, \sigma_\eta^2, \sigma_\varsigma^2, \mu_0)$ and $\boldsymbol{x}_1^T$ where $\boldsymbol{x}_1^T = (x_1, x_2, ..., x_T)$.

$$B = -\frac{\ln k}{\Delta t} \tag{2.52}$$

$$\gamma = \sigma_\eta \sqrt{\frac{2B}{1 - e^{-2B\Delta t}}} \tag{2.53}$$

Inference was made using the joint probability of the observed ($\boldsymbol{y}_1^T$) and unobserved ($\boldsymbol{x}_1^T$) variables. The joint probability listed as (2.54) came from (2.51) and the convenient Markov properties of the DLM. Using (2.54), inference was made using two methods. The first was a variant of the EM method described in Section 2.1 and point estimates for $\hat{k}$, $\hat{\sigma}_\eta^2$, $\hat{\sigma}_\varsigma^2$,$\hat{\mu}_0$, and $\hat{\boldsymbol{x}}_1^T$ were made. The second was a Bayesian model as described in Section 2.2 that simulates the posterior distribution of k, σ_η^2, σ_ς^2, μ_0, and $\boldsymbol{x}_1^T$ using a Gibbs sampler.

$$p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T | \boldsymbol{\theta}) = p(x_1, y_1 | \boldsymbol{x}_2^T, \mu_0, \sigma_\eta^2) \prod_{i=2}^{T} \left[p(x_i | \boldsymbol{x}_0^{i-1}, \boldsymbol{x}_{i+1}^T, k, \sigma_\eta^2) p(y_i | x_i, \sigma_\varsigma^2) \right]$$

$$p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T | \boldsymbol{\theta}) = p(x_1 | \mu_1, \sigma_\eta^2) p(y_1 | x_1, \sigma_\varsigma^2) \prod_{i=2}^{T} \left[p(x_i | x_{i-1}, k, \sigma_\eta^2) p(y_i | x_i, \sigma_\varsigma^2) \right] \tag{2.54}$$

2.3.3 Estimating the parameters of a DLM using the EM method

For the EM method, the expectation step and maximization step must be computed. For this, consider the log of the joint probability that was written as (2.54) where $\boldsymbol{\theta} = (k, \sigma_\eta^2, \sigma_\varsigma^2, \mu_0)$.

$$\begin{aligned}
\ln(p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\boldsymbol{\theta})) &= \ln\{p(x_1|\boldsymbol{\theta})\} + \sum_{i=2}^{T} \ln\{p(x_i|x_{i-1}, \boldsymbol{\theta})\} + \sum_{i=1}^{T} \ln\{p(y_i|x_i, \boldsymbol{\theta})\} \\
\ln\{p(x_1|\boldsymbol{\theta})\} &= -\frac{1}{2}\ln(2\pi\sigma_\eta^2) - \frac{1}{2\sigma_\eta^2}(x_1 - \mu_0)^2 \\
\sum_{i=2}^{T} \ln\{p(x_i|x_{i-1}, \boldsymbol{\theta})\} &= -\frac{T-1}{2}\ln(2\pi\sigma_\eta^2) - \sum_{i=2}^{T}\frac{1}{2\sigma_\eta^2}(x_i^2 - 2kx_ix_{i-1} + k^2x_{i-1}^2) \\
\sum_{i=1}^{T} \ln\{p(y_i|x_i, \boldsymbol{\theta})\} &= -\frac{T}{2}\ln(2\pi\sigma_\varsigma^2) - \sum_{i=1}^{T}\frac{1}{2\sigma_\varsigma^2}(y_i^2 - 2y_ix_i + x_i^2)
\end{aligned}$$

For the j^{th} expectation step, $\mathbb{E}_{\boldsymbol{x}_1^T|\hat{\boldsymbol{\theta}}_{j-1}, \boldsymbol{y}_1^T}[\ln(p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\boldsymbol{\theta}))]$ was computed. For this, it was sufficient to compute $\mathbb{E}_{\boldsymbol{x}_1^T|\hat{\boldsymbol{\theta}}_{j-1}, \boldsymbol{y}_1^T}[x_i]$, $\mathbb{E}_{\boldsymbol{x}_1^T|\hat{\boldsymbol{\theta}}_{j-1}, \boldsymbol{y}_1^T}[x_i^2]$, and $\mathbb{E}_{\boldsymbol{x}_1^T|\hat{\boldsymbol{\theta}}_{j-1}, \boldsymbol{y}_1^T}[x_ix_{i-1}]$. For brevity, they were denoted as $(\hat{x}_i)_j$, $\left(\widehat{x_i^2}\right)_j$, and $(\widehat{x_{i-1}x_i})_j$.

The E-step: Computing $(\hat{x}_i)_j$, $\left(\widehat{x_i^2}\right)_j$, and $\left(\widehat{x_{i-1}x_i}\right)_j$

To compute $(\hat{x}_i)_j$, $\left(\widehat{x_i^2}\right)_j$, and $(\widehat{x_{i-1}x_i})_j$, $(\hat{x}_i)_j$ and $\left(\widehat{x_i^2}\right)_j - (\hat{x}_i^2)_j$ were computed. Although $\left(\widehat{x_i^2}\right)_j - (\hat{x}_i^2)_j$ which was denoted as $\left(s_{x_i}^2\right)_j$ was not a quantity of interest, it was computed as it was necessary to find $\left(\widehat{x_i^2}\right)_j$, and $(\widehat{x_{i-1}x_i})_j$. $(\hat{x}_i)_j$ and $\left(s_{x_i}^2\right)_j$ were computed using a Kalman Filter which was developed by Rudolph Kalman in 1960 [26]. The Kalman Filter, which is a special case of the forward-backwards method, consists of computing the expectation and variance of three probabilities. The first two of these are the state forecast $(p(x_i|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta}))$ and the state update $(p(x_i|\boldsymbol{y}_1^{i}, \boldsymbol{\theta}))$ which were denoted as a_i and α_i respectively.

$$\hat{a}_i = \mathbb{E}(a_i) \qquad\qquad s^2_{a_i} = \mathrm{Var}(a_i) \tag{2.55}$$

$$\hat{\alpha}_i = \mathbb{E}(\alpha) \qquad\qquad s^2_{\alpha_i} = \mathrm{Var}(\alpha_i) \tag{2.56}$$

The expectation and variance of each a_i and α_i were computed recursively from $i = 1$ to T. They were denoted as (2.55) and (2.56) and the values associated were listed in (2.57). For the purpose of parameter estimation, the expectation and variance of $x_i|\boldsymbol{y}_0^T, \hat{\boldsymbol{\theta}}_{j-1}$ was needed. However, Kalman Filters can also be used for state estimation and prediction when $\boldsymbol{\theta}$ was known. Therefore, subscripts of the expectation, variance, and $\boldsymbol{\theta}$ were left off to maintain generality; but when implementing the entirety of the EM algorithm from this section, all $\boldsymbol{\theta}$ should be $\hat{\boldsymbol{\theta}}_{j-1}$.

Computing the State Forecast and State Updates (2.57)

```
I   $\hat{a}_1 = \mu_0$
II  $s^2_{a_1} = \sigma^2_\eta$
III $\hat{\alpha}_1 = \hat{a}_1 + \frac{s^2_{a_1}}{\sigma^2_\varsigma}(y_1 - \hat{a}_1)$
IV  $s^2_{\alpha_1} = \frac{\sigma^2_\varsigma s^2_{a_1}}{\sigma^2_\varsigma + k^2 s^2_{a_1}}$
V   For $i = 1 : T$
    (a) $\hat{a}_i = k\hat{\alpha}_{i-1}$
    (b) $s^2_{a_i} = \sigma^2_\eta + k^2 s^2_{\alpha_{i-1}}$
    (c) $s^2_{\alpha_i} = \frac{\sigma^2_\varsigma s^2_{a_i}}{\sigma^2_\varsigma + k^2 s^2_{a_i}}$
    (d) $\hat{\alpha}_i = \hat{a}_i + \frac{s^2_{\alpha_i}}{\sigma^2_\varsigma}(y_i - \hat{a}_i)$
VI  end
```

The backwards probability was defined as $p(x_i|\boldsymbol{y}_0^T, \boldsymbol{\theta})$ or the probability of x_i given the full data. Following that, the expectation and variance of the backward probability were already defined as $(\hat{x}_i)_j$ and $\left(s_{x_i}^2\right)_j$. As with the forward portion, the backward portion was computed recursively. The algorithm for computing the backward probability was listed as (2.58).

Computing the expectation and variance given the full data (2.58)

```
I $\hat{x}_T = \hat{\alpha}_T$
II $s_{x_T}^2 = s_{\alpha_T}^2$
III For $i = T - 1 : 1$
    (a) $\hat{x}_i = \hat{\alpha}_i + \frac{s_{\alpha_i}^2}{s_{a_{i+1}}^2} \left[\hat{x}_{i+1} - \hat{a}_{i+1}\right]$
    (b) $s_{x_i}^2 = s_{\alpha_i}^2 + \left(\frac{k s_{\alpha_i}^2}{s_{\alpha_{i+1}}^2}\right)^2 \left[s_{x_{i+1}}^2 - s_{a_{i+1}}^2\right]$
IV end
```

Then, $\left(\widehat{x_i^2}\right)_j$, and $\left(\widehat{x_{i-1} x_i}\right)_j$ could be computed directly from the results of the forward and backward iterations. They were listed as (2.59) and (2.60) respectively.

$$\widehat{x_i^2} = s_{x_i}^2 + (\hat{x}_i)^2 \tag{2.59}$$

$$\widehat{x_{i-1} x_i} = \hat{x}_{i+1}^2 \hat{\alpha}_i + \frac{k s_{\alpha_i}^2}{s_{a_{i+1}}^2} \left[\widehat{x_i^2} - \hat{a}_{i+1} \hat{x}_{i+1}\right] \tag{2.60}$$

A derivation of the E-step can be found in Appendix B.2.1. In addition, derivations can be found in [54] and [15]. [15] is more explicit and is a good source to build an initial understanding of the forward-backward method. [54] is a comprehensive book on time series and provides information on the complete EM method.

The M-step

For the j^{th} M-step, it was necessary to find the $\boldsymbol{\theta}_j$ that maximized $\mathbb{E}_{\boldsymbol{x}_1^T|\hat{\boldsymbol{\theta}}_{j-1},\boldsymbol{y}_1^T}\Big[$ $\ln\left(p\left(\boldsymbol{x}_1^T,\ \ \boldsymbol{y}_1^T|\boldsymbol{\theta}\right)\right)\Big]$. The maximizing values of $\boldsymbol{\theta}_j$ given $(\hat{x}_i)_j$, $\left(\widehat{x_i^2}\right)_j$, and $\left(\widehat{x_{i-1}x_i}\right)_j$ were listed in (2.61) to (2.64).

$$(\hat{\mu}_0) = (\hat{x}_1)_j \tag{2.61}$$

$$\hat{k}_j = \frac{\sum_{i=2}^{T}\left(\widehat{x_{i-1}x_i}\right)_j}{\left(\widehat{x_i^2}\right)_j} \tag{2.62}$$

$$MSE\left((\hat{x}_i)_j, \hat{k}_j\,(\hat{x}_{i-1})_j\right) = \sum_{i=2}^{T}\left(\left(\widehat{x_i^2}\right)_j - \hat{k}_j\left(\widehat{x_{i-1}x_i}\right)_j + (\hat{k}_j)^2\left(\widehat{x_{i-1}^2}\right)_j\right)$$

$$MSE\left((\hat{x}_1)_j, (\hat{\mu}_0)_j\right) = \left(\left(\widehat{x_1^2}\right)_j - 2\,(\hat{\mu}_0)_j\,(\hat{x}_1)_j + \left((\hat{\mu}_0)_j\right)^2\right)$$

$$\left(\hat{\sigma}_\eta^2\right)_j = \frac{MSE\left((\hat{x}_i)_j, \hat{k}_j\,(\hat{x}_{i-1})_j\right) + MSE\left((\hat{x}_1)_j, (\hat{\mu}_0)_j\right)}{T} \tag{2.63}$$

$$\left(\hat{\sigma}_\varsigma^2\right)_j = \frac{1}{T}\left[\sum_{i=1}^{T}\left(y_i^2 - 2y_i\,(\hat{x}_i) + \left(\widehat{x_i^2}\right)_j\right)\right] \tag{2.64}$$

The EM-algorthm for a DLM

The following describes the full EM-algorithm for the DLM, where δ was the arbitrary stopping condition. For further information on the full algorithm, see [54]

I Pick $\hat{\boldsymbol{\theta}}_0$

II Set $j = 0$ and define $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$.

III While $\left|\left|\hat{\boldsymbol{\theta}}_j - \hat{\boldsymbol{\theta}}_{j-1}\right|\right|_\infty > \delta$

(a) $j = j + 1$

(b) Calculate all $(\hat{a}_i)_j$, $\left(s_{a_i}^2\right)_j$, $(\hat{\alpha}_i)_j$, and $\left(s_{\alpha_i}^2\right)_j$ using (2.57)

(c) Calculate all $(\hat{x}_i)_j$ and $\left(s_{x_i}^2\right)_j$ using (2.58).

(d) Calculate $\left(\widehat{x_1^2}\right)_j$ using (2.59).

(e) For $i = 2 : T$

i. Calculate $\left(\widehat{x_i^2}\right)_j$ using (2.59).

ii. Calculate $\left(\widehat{x_{i-1}x_i}\right)_j$ using (2.60).

(f) Calculate $(\hat{\mu}_0)_j$ using (2.61).

(g) Calculate $\hat{k}_j$ using (2.62).

(h) Calculate $\left(\sigma_\eta^2\right)_j$ using (2.63).

(i) Calculate $\left(\sigma_\varsigma^2\right)_j$ using (2.64).

2.3.4 Estimating the parameters of a DLM using Bayesian Inference

For the Bayesian model, σ_η^2, σ_ς^2, k, and μ_0 were treated as random variables. The joint posterior of $\boldsymbol{\theta}$ was intractable, so a Gibbs sampler was used. Like the EM method, information on $\boldsymbol{x}_1^T|\boldsymbol{y}_0^T$ was required to sample from the posterior of $\boldsymbol{\theta}$. Therefore, the conditional posterior of $\boldsymbol{x}_1^T$ was also sampled using a method called the forward filtering backward sampling (FFBS) [59] [16].

Forward filter-backward sample

To sample the conditional posterior of $\boldsymbol{x}_1^T$, it was necessary to draw from the distribution of $p(x_i|\boldsymbol{y}_1^T, \boldsymbol{\theta})$. For this, the expectation and variance of the state forecast and state update were needed. Following this, the forward filter was computed using (2.57) exactly as with the EM method. Then $\left(\boldsymbol{x}_0^T\right)_j$ was drawn from the the conditional posterior using (2.65) where c_j denotes the j^{th} draw from the conditional posterior of c and $(c_s)_j$ denotes the same for a sub-scripted random variable.

The backward sample for a DLM (2.65)

I $(x_T)_j = \mathcal{N}_{x_T}\left((\hat{\alpha}_T)_j, \left(s_{\alpha_T}^2\right)_j\right)$

II For $i = T-1:1$

   (a) Draw $\left(x_i\right)_j \sim \mathcal{N}_{x_i}\left(\hat{\alpha}_i + \frac{\left(s^2_{\alpha_i}\right)_j \hat{k}_{j-1}}{\left(s^2_{a_{i+1}}\right)_j}\left[\left(x_{t+1}\right)_j - \hat{a}_{i+1}\right], \frac{\left(\sigma^2_\eta\right)_{j-1}\left(s^2_{\alpha_i}\right)_j}{\left(s^2_{a_{i+1}}\right)_j}\right)$

III end

The derivation follows Appendix (B.2.1) through Appendix (B.12). Since $\left(x_i\right)_j$ was drawn in reverse order, $\left(x_{i+1}\right)_j$ had already been drawn when $\left(x_i\right)_j$ was drawn. Then, the current conditional posterior of $\left(x_{i+1}\right)_j$ was used in drawing $\left(x_i\right)_j$. This made expectation and variance calculated for the Kalman Filter in Appendix (B.2.1) after (B.12) not necessary. Therefore, for the Bayesian Model all information in Appendix (B.2.1) after (B.12) can be ignored.

$$
\begin{aligned}
p(\sigma^2_\eta | \boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\sigma^2_\eta}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T | \sigma^2_\varsigma, \sigma^2_\eta, k, \mu_0) p(\sigma^2_\eta) \\
p(\sigma^2_\varsigma | \boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\sigma^2_\varsigma}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T | \sigma^2_\varsigma, \sigma^2_\eta, k, \mu_0) p(\sigma^2_\varsigma) \\
\sigma^2 &\sim \mathcal{IG}\left(\frac{n}{2}, \frac{d}{2}\right) \\
\sigma^2_\eta | \boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\sigma^2_\eta} &\sim \mathcal{IG}\left(\frac{T+n_\eta}{2}, \frac{\left[d_\eta + (x_1-\mu_0)^2 + \sum\limits_{i=2}^{T}(x_i - kx_{i-1})^2\right]}{2}\right) \qquad (2.66) \\
\sigma^2_\varsigma | \boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\sigma^2_\varsigma} &\sim \mathcal{IG}\left(\frac{T+n_\varsigma}{2}, \frac{1}{2}\left[d_\varsigma + \sum_{i=1}^{T}(y_i - x_i)^2\right]\right) \qquad (2.67)
\end{aligned}
$$

A derivation for (2.66) and (2.67) can be found in Appendix B.2.3.1 and B.2.3.2 respectively. If there is little or no prior information about σ^2_η and σ^2_ς, the flat priors $p(\sigma^2_\eta) \propto \frac{1}{\sigma^2_\eta}$ and $p(\sigma^2_\eta) \propto \frac{1}{\sigma^2_\varsigma}$ can be employed. The resulting posteriors are the same as (2.66) and (2.67) except $n_\eta = d_\eta = n_\varsigma = d_\varsigma = 0$.

For, k, the conditionally conjugate prior was not used. The normal distribution is the conditionally conjugate prior for k. There are two possible problems with this choice of prior. The first is the effect of the prior on the posterior. This can be minimized by picking large values for the variance of the normal distribution. The second problem is that this prior can draw values of k such that $k > 1$ or $k < 0$. In the case $k > 1$ or $k < 0$, it is expected that x_i goes to infinity, which was not the case being considered. Therefore $p(k)$ must restricted from 0 to 1. A prior that achieves this and results in a conditional posterior that could be sampled was the uniform prior from 0 to 1. The conditional posterior was listed as (2.68) and $I[0 \leq k \leq 1]$ denotes the indicator function.

$$\begin{aligned} p(k|\boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-k}) &\propto p(\boldsymbol{y}_1^T, \boldsymbol{x}_1^T|k, \mu_0, \sigma_\eta^2, \sigma_\varsigma^2)p(k) \\ p(k) &= I[0 \leq k \leq 1] \\ k|\boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-k} &\sim \mathcal{TN}\left(\frac{\sum_{i=2}^T x_i x_{i-1}}{\sum_{i=2}^T x_{i-1}^2}, \frac{\sigma_\eta^2}{\sum_{i=2}^T x_{i-1}^2}; (0,1)\right) \end{aligned} \tag{2.68}$$

The posterior in (2.68) is proportional to a truncated normal on $(0, 1)$ with mean $\frac{\sum\limits_{i=2}^{T} x_i x_{i-1}}{\sum\limits_{i=2}^{T} x_{i-1}^2}$, variance $\frac{\sigma_\eta^2}{\sum\limits_{i=2}^{T} x_{i-1}^2}$. A derivation for (2.68) can be found in Appendix B.2.3.3.

The conditionally conjugate prior of μ_0 was also a normal distribution. For μ_0, the conjugate prior was used since there was a little more information. Since μ_0 represents an initial state before x_1, and since each state is normally distributed, it is logical to place a normal prior for μ_0 near y_1. However, since there is no direct information available for this initial state, it was advantageous to make the prior weak. As before, the normal prior can be made weak with a large variance. Therefore, the prior $\mu_0 \sim \mathcal{N}\left(m_0, \sigma_m^2\right)$ was used with $\mu_0 \approx y_1$ and σ_m^2 large. The resulting posterior was listed as (2.69) and a derivation can be found in Appendix B.2.3.4.

$$\begin{aligned} p(\mu_0|\boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\mu_0}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\sigma_\varsigma^2, \sigma_\eta^2, k, \mu_0)p(\mu_0) \\ \mu_0|\boldsymbol{y}_1^T, \boldsymbol{x}_1^T, \boldsymbol{\theta}_{-\mu_0} &\sim N\left(\frac{x_1\sigma_m^2 + m_0\sigma_\eta^2}{\sigma_m^2 + \sigma_\eta^2}, \frac{\sigma_\eta^2\sigma_m^2}{\sigma_m^2 + \sigma_\eta^2}\right) \end{aligned} \tag{2.69}$$

The Gibbs Sampler

To create an accurate representation of the posterior, N draws were taken from each conditional posterior. At the beginning of the simulation initial values for $\boldsymbol{\theta}_0$ must be chosen as with the EM algorithm. With the initial values set, draws are taken from each of the following distributions.

I Choose $\boldsymbol{\theta}_0$

II For $j = 1 : N$

(a) Draw $\left(\boldsymbol{x}_1^T\right)_j$ from $p\left(\boldsymbol{x}_1^T \middle| \boldsymbol{y}_1^T \left(\sigma_\eta^2\right)_{j-1}, \left(\sigma_\varsigma^2\right)_{j-1}, k_{j-1}, \left(\mu_0\right)_{j-1}\right)$ listed as (2.65).

(b) Draw $\left(\sigma_\eta^2\right)_j$ from $p\left(\sigma_\varsigma^2 \middle| \left(\boldsymbol{x}_1^T\right)_j, \boldsymbol{y}_1^T, \left(\sigma_\eta^2\right)_{j-1}, k_{j-1}, \left(\mu_0\right)_{j-1}\right)$ listed as (2.66).

(c) Draw $\left(\sigma_\varsigma^2\right)_j$ from $p\left(\sigma_\eta^2 \middle| \left(\boldsymbol{x}_1^T\right)_j, \boldsymbol{y}_1^T, \left(\sigma_\varsigma^2\right)_{j-1}, k_{j-1}, \left(\mu_0\right)_{j-1}\right)$ listed as (2.67).

(d) Draw $(k)_j$ from $p\left(k \middle| \left(\boldsymbol{x}_1^T\right)_j, \boldsymbol{y}_1^T, \left(\sigma_\varsigma^2\right)_{j-1}, \left(\sigma_\eta^2\right)_{j-1}, \left(\mu_0\right)_{j-1}\right)$ listed as (2.68).

(e) Draw $\left(\mu_0\right)_j$ from $p\left(\mu_0 \middle| \left(\boldsymbol{x}_1^T\right)_j, \boldsymbol{y}_1^T, \left(\sigma_\varsigma^2\right)_{j-1}, \left(\sigma_\eta^2\right)_{j-1}, k_{j-1}\right)$ listed as (2.69).

III end

The DLM discussed was one special case of the larger family of DLMs. For more information on Bayesian inference on DLMs see [59] and for more on Bayesian inference on general time series see [41].

2.3.5 Studying the effect of data collection on the inference of B, γ, and σ_ς^2

The test function (2.70)

$$
\begin{aligned}
dX &= -BXdt + \gamma dW \\
B &= \gamma = 1 \\
y_i &= X(t_i) + \varsigma & \varsigma \sim \mathcal{N}_\varsigma(0, \sigma_\varsigma^2)
\end{aligned}
$$

The system to be inferred on was listed as 2.70. B and γ were the primary interest, but σ_ς^2 was necessary to make inference on the aforementioned parameters and it also quantified noise from the data collection. The inference on B and γ varies depending on the values of k, σ_η^2, σ_ς^2, and the number of observations. The values of k, σ_η^2, σ_ς^2, and the number of observations are determined by the choice of Δt and total time observed. Therefore, the effect of Δt and the total time observed $(t_T - t_1)$ on the quality of inference was studied.

To do this, a few different experiments were run. For all inference, Bayesian posteriors were used to evaluate as much more could be extrapolated from a single simulation and the associated inference. Since k varies from $(0, 1)$ regardless of choice of parameter in (2.70) and σ_η^2 varies from $\left(0, \frac{\gamma^2}{2B}\right)$ than $B = \gamma = 1$ was chosen for all experiments. The values of k and σ_η^2 given $B = \gamma = 1$ were shown as functions of Δt in Figure 2.1. The first experiment (2.3.5) studied inference while the number of observations of the DLM was constant and Δt was varied. The second (2.3.5) studied inference while Δt was constant and $t_F - t_1$ was varied. Finally, the last experiment (2.3.5) holds $t_F - t_1$ constant as the number of observations and Δt were varied.

Varying $\Delta t = 8$ to $\Delta t = \frac{1}{16384}$ for Dynamic Linear Models with $10,000$ observations

For each trial in this experiment, there are $10,000$ observations. Δt was varied from $\Delta t = 8$ to $\Delta t = \frac{1}{16384}$ and Δt for each successive trial was half the previous trial.

Since the number of observations was constant, the total elapsed time $(t_F - t_1)$ was

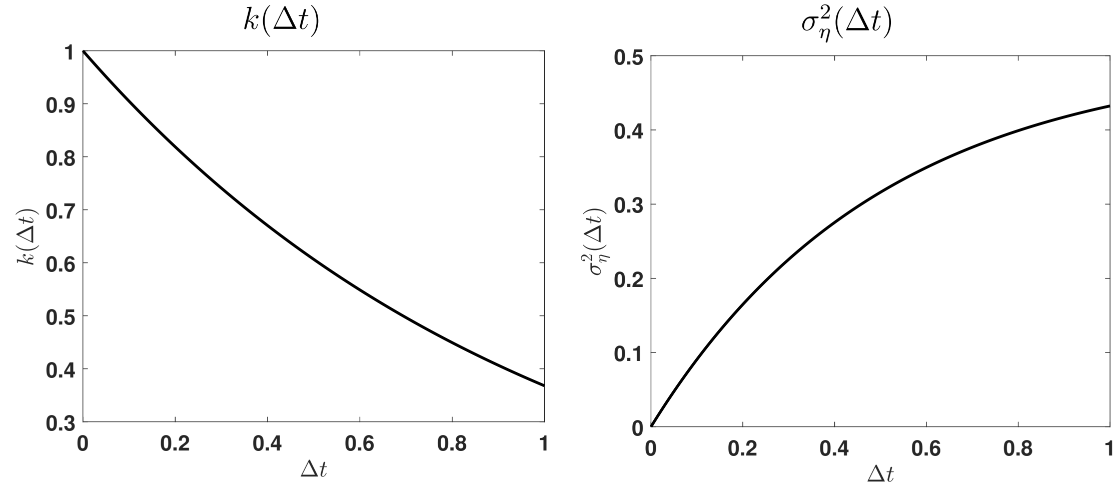


Figure 2.1: The value of $k(\Delta t)$ and $\sigma_\eta(\Delta t)^2$ for $\Delta t \in [0, 1]$

also half of the previous trial. Given that, each trial was a separately simulated dynamic linear models.

To explore inference for each Δt, let $(\epsilon_c)_j$ be the difference between the j^{th} draw from the posterior and the true value of c or $(\epsilon_c)_j + c_j = c$. The distributions were compared by plotting the credible intervals of $\left\{(\epsilon_c)_j\right\}$ where $\left\{(\epsilon_c)_j\right\}$ was the set of all $(\epsilon_c)_j$ for each trial on the same axis. 98%, 95%, and 80% CIs (Credible Intervals) were demarcated by the color schemes in Figure 2.2. Unfortunately, the outliers made it difficult to observe the trends of the credible intervals from $\Delta t = 8$ to $\Delta t = \frac{1}{16384}$ for $\left\{(\epsilon_B)_j\right\}$ and $\left\{(\epsilon_\gamma)_j\right\}$. Therefore, the transformation $g\left(\left\{(\epsilon_c)_j\right\}\right)$ where $g(x) = \sinh^{-1}\left(\sinh^{-1}(x)\right)$ were plotted in Figures 2.3 and 2.4 for $c = B$ and $c = \gamma$ respectively. Each trial was labeled on the

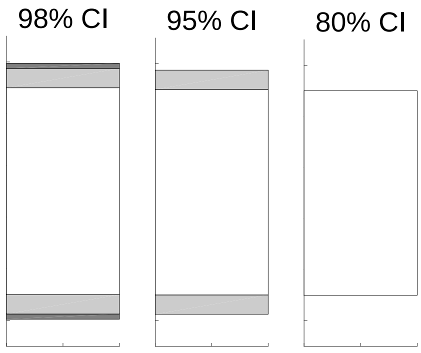


Figure 2.2: CI Demarcation

x-axis with their corresponding Δt. Finally, $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ had a slightly different structure for $\Delta t = 8$ to $\Delta t = \frac{1}{16384}$. Following this, the transformation used for $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ was $h\left(\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}\right)$ where $h(x) = g(50x)$. $h\left(\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}\right)$ was plotted in Figure 2.5.

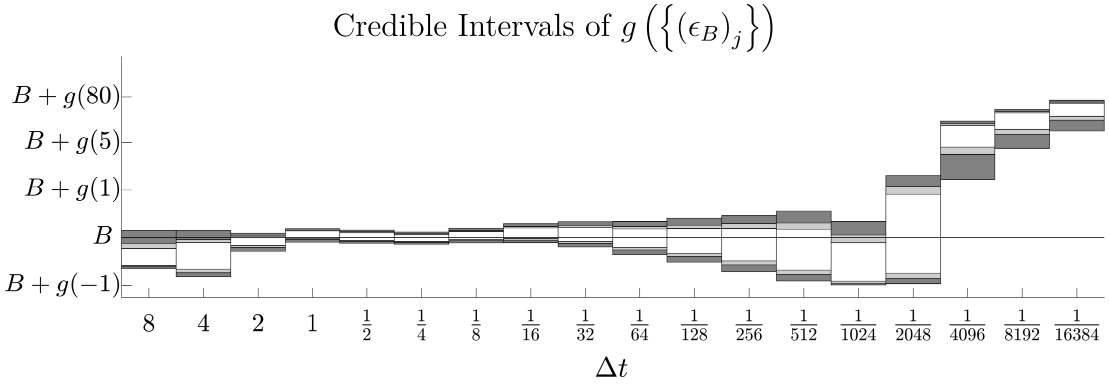


Figure 2.3: $g\left(\left\{\left(\epsilon_B\right)_j\right\}\right)$ for experiment 2.3.5

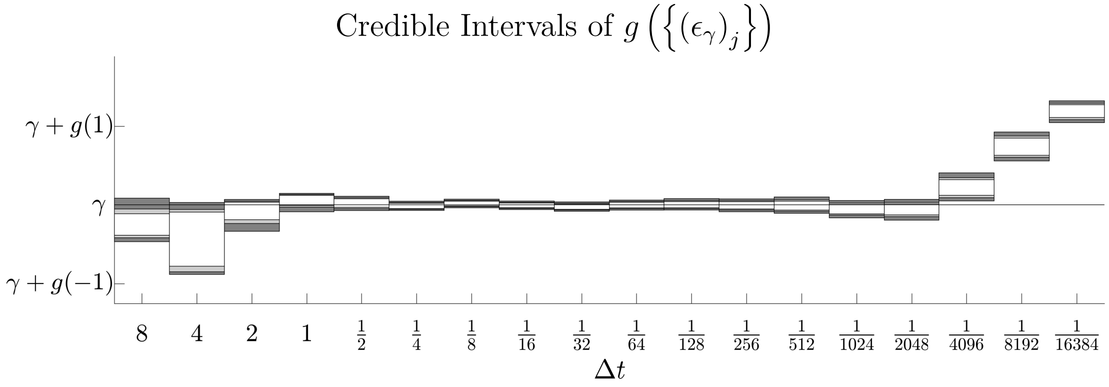


Figure 2.4: $g\left(\left\{\left(\epsilon_\gamma\right)_j\right\}\right)$ for experiment 2.3.5

Examining Figures 2.3, 2.4, and 2.5 revealed three distinct region of behaviors. For large Δt, it was seen the credible intervals of $\left\{\left(\epsilon_\gamma\right)_j\right\}$ and $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ were large. For small Δt the credible intervals of $\left\{\left(\epsilon_B\right)_j\right\}$ and $\left\{\left(\epsilon_\gamma\right)_j\right\}$ were large and do not appear to be centered around the true values of B and γ. Finally, for intermittent values of Δt, the credible intervals remained more stable either growing or shrinking slightly for each change in Δt. Since these regions displayed distinct behaviors, each was investigated

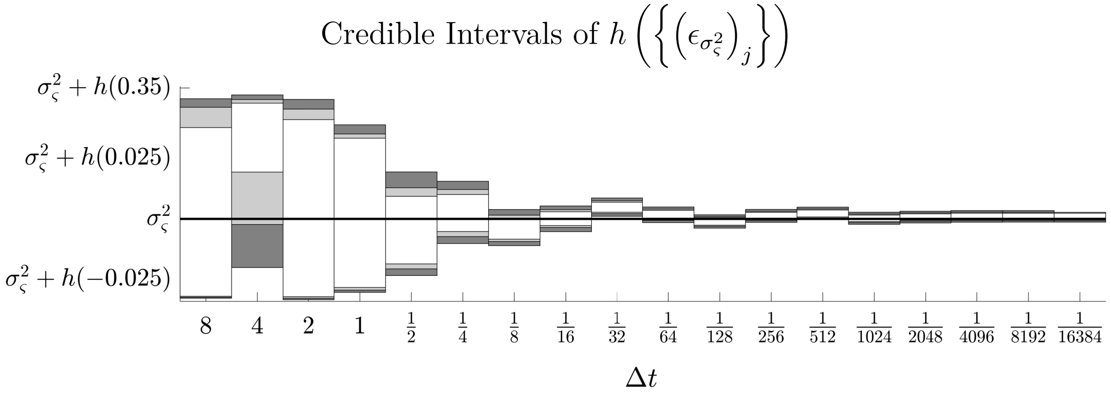


Figure 2.5: $h\left(\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}\right)$ for experiment 2.3.5

separately.

First, $\Delta t = 8$ to $\Delta t = 1$ was explored. Figure 2.6 has the histograms of $\{B_j\}$, $\{\gamma_j\}$, and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for $\Delta t = 8$. It is clear that distributions of $\{\gamma_j\}$ and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ are not good estimates of γ and σ_ς^2. To better explore this phenomenon $\{k_j\}$ and $\left\{\left(\sigma_\eta^2\right)_j\right\}$ were also placed in Figure 2.6 since the *MCMC* algorithm takes draws from the posteriors of k, σ_η^2, and σ_ς^2.

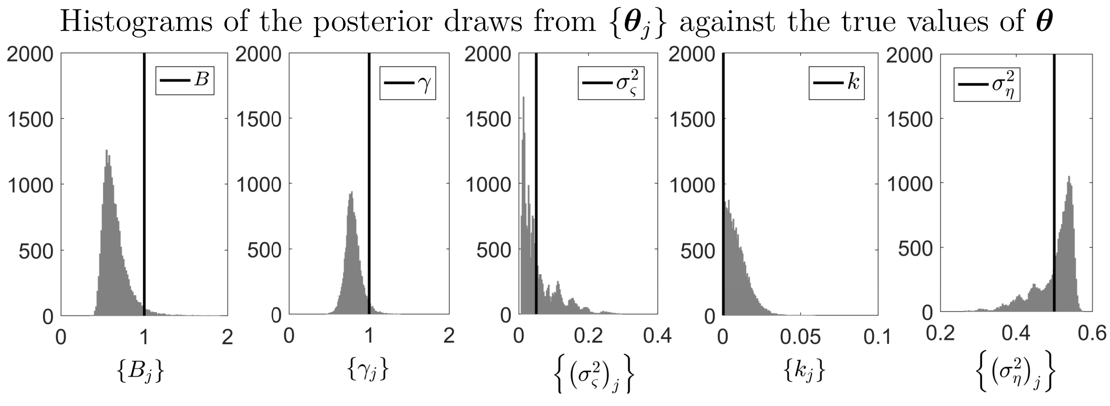


Figure 2.6: Histograms of $\{B_j\}$, $\{\gamma_j\}$, $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$, $\{k_j\}$, and $\left\{\left(\sigma_\eta^2\right)_j\right\}$ for $\Delta t = 8$ of experiment 2.3.5

Recall k, σ_η^2, and σ_ς^2 were defined by (2.51). If $k = 0$ the system was reduced to $x_i = \sigma_\eta^2$ and $y_t = x_i + \sigma_\varsigma^2$ which can be simplified to $y_i = \sigma_\eta^2 + \sigma_\varsigma^2$. Since x_i was unobserved, it was impossible to discern between σ_η^2 and σ_ς^2 in the case where $k = 0$. It is interesting to note, for this simulation $\sigma_\eta^2 + \sigma_\varsigma^2 = 0.55$ and both σ_η^2 and σ_ς^2 explore a great deal of the space from 0 to 0.55. Following this, the posteriors of σ_η^2 and σ_ς^2 did not perform well for low values of k. In Figures 2.7 and 2.8, the identifiability issues persists from $\Delta t = 4$ to $\Delta t = 1$ and starts rapidly improving in $\Delta t = \frac{1}{2}$ and $\frac{1}{4}$.

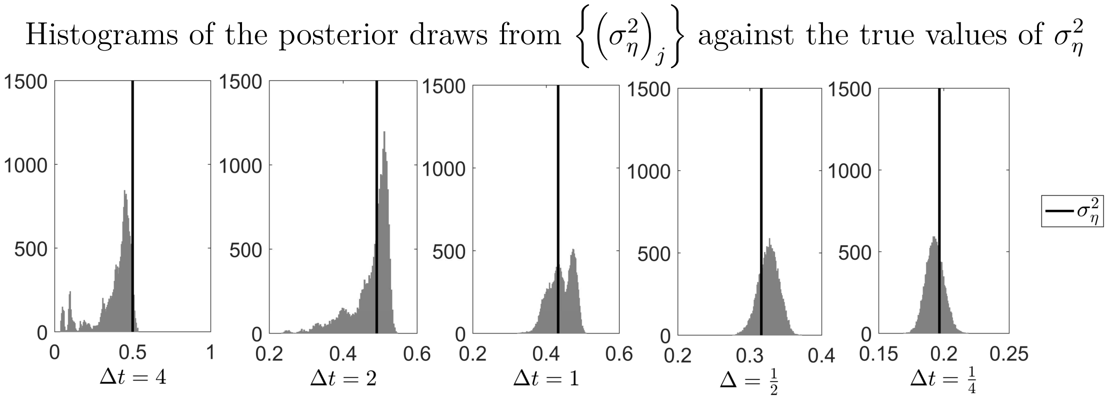


Figure 2.7: Histograms of $\left\{\left(\sigma_\eta^2\right)_j\right\}$ for $\Delta t = 4$ to $\Delta t = \frac{1}{4}$ of experiment 2.3.5

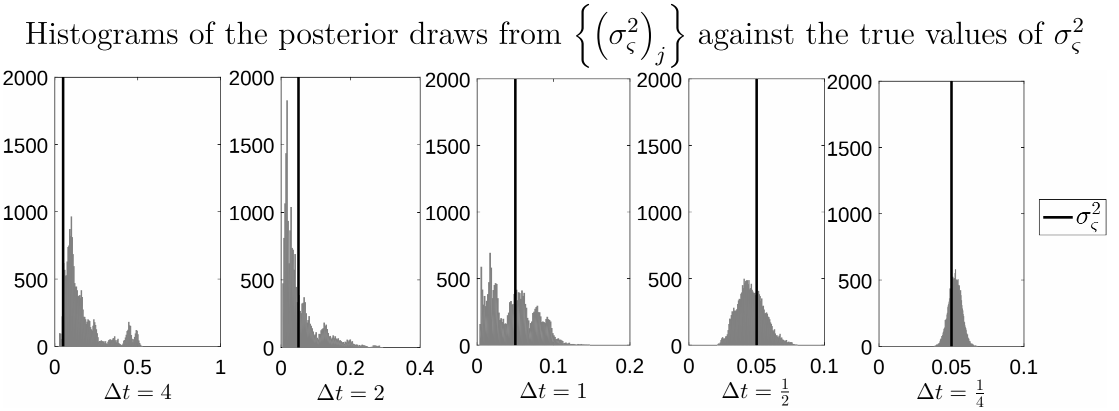


Figure 2.8: Histograms of $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for $\Delta t = 4$ to $\Delta t = \frac{1}{4}$ of experiment 2.3.5

The inference on B, γ, and σ_ς^2 exhibit more positive behavior for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$. The credible intervals did not exhibit anomalies and are centered around the true values of B, γ, and σ_ς^2. Figures 2.9, 2.10, and 2.11 compare the credible intervals of $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ using the same visualization as Figure 2.2.

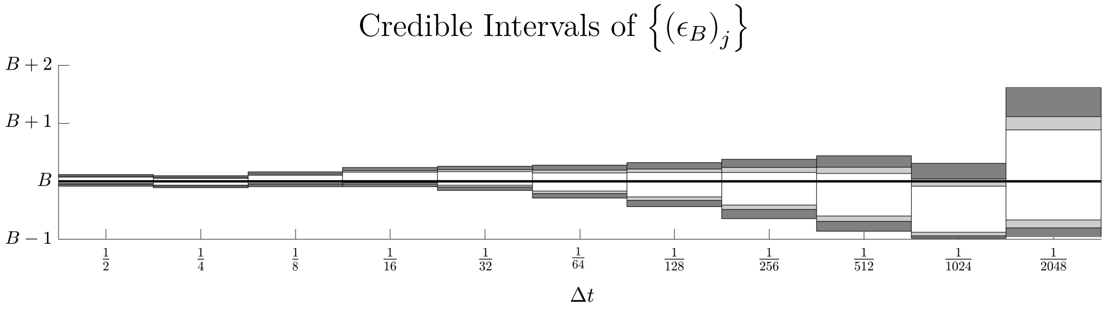


Figure 2.9: $\left\{(\epsilon_B)_j\right\}$ for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ of experiment 2.3.5

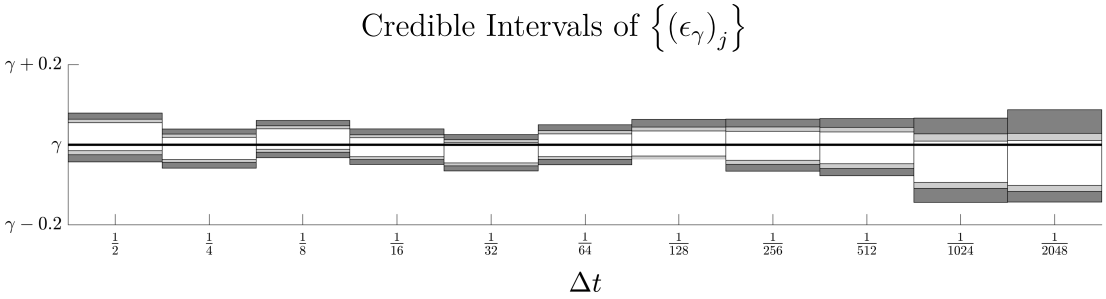


Figure 2.10: $\left\{(\epsilon_\gamma)_j\right\}$ for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ of experiment 2.3.5

Only the CI for $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ became more precise as Δt got smaller. As before, $\left\{(\epsilon_k)_j\right\}$ and $\left\{\left(\epsilon_{\sigma_\eta^2}\right)_j\right\}$ were investigated to understand this behavior. The credible intervals of $\left\{(\epsilon_k)_j\right\}$ and $\left\{\left(\epsilon_{\sigma_\eta^2}\right)_j\right\}$ were plotted in Figures 2.12 and 2.13.

One can observe the credible intervals of $\left\{(\epsilon_k)_j\right\}$ and $\left\{\left(\epsilon_{\sigma_\eta^2}\right)_j\right\}$ are decreasing in size for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$. Therefore, in these two cases the lack of improvement as Δt gets small for $\left\{(\epsilon_B)_j\right\}$ and $\left\{(\epsilon_\gamma)_j\right\}$ must exist in the transformation from k and σ_η^2 to B and γ. Since increase of the size of the credible intervals of $\left\{(\epsilon_B)_j\right\}$ was more

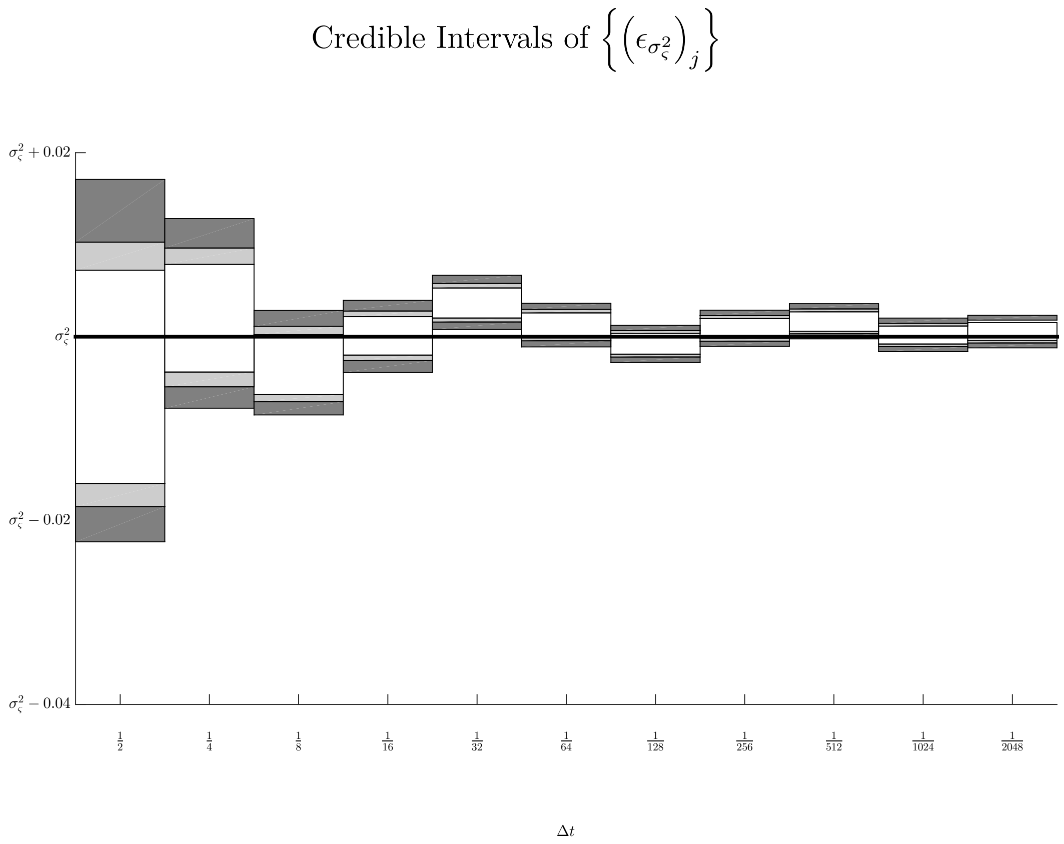


Figure 2.11: $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ of experiment 2.3.5

extreme, the transformation from k to B appears to be the limiting factor. Therefore, the transformation from k to B was studied.

$$B = -\frac{\ln k}{\Delta t} \tag{2.71}$$

Then behavior of $(\epsilon_B)_j$ was studied using an expansion around the true value of k and a derivation of the expansion can be found in Appendix B.2.3.5.

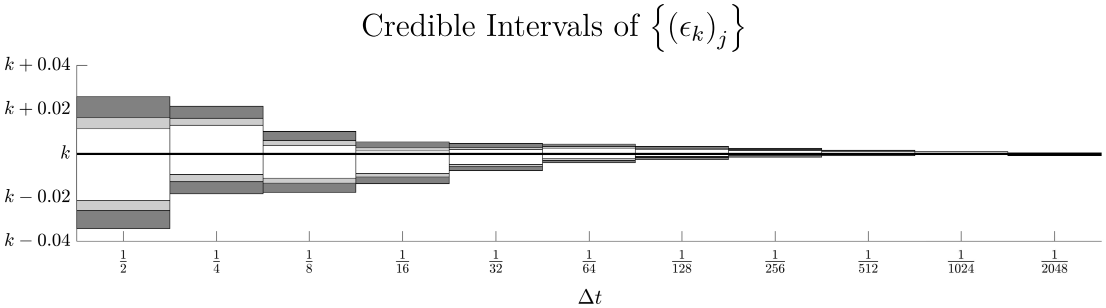


Figure 2.12: $\left\{(\epsilon_k)_j\right\}$ for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ of experiment 2.3.5

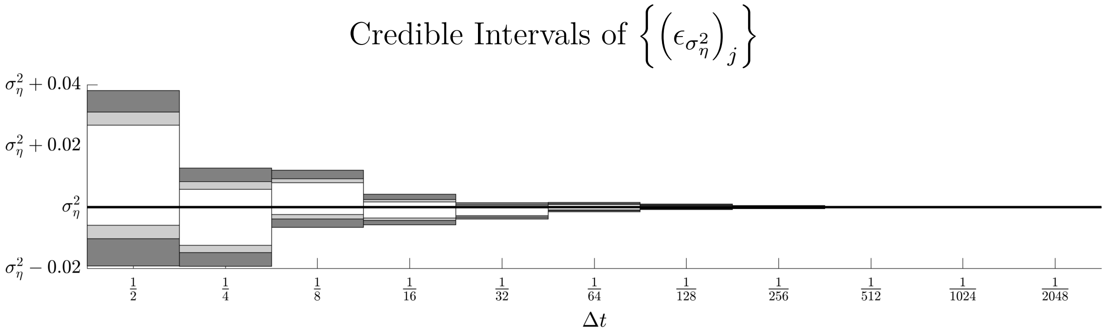


Figure 2.13: $\left\{\left(\epsilon_{\sigma_\eta^2}\right)_j\right\}$ for $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{2048}$ of experiment 2.3.5

$$\begin{aligned}
B_j &= -\frac{\ln(k_j)}{\Delta t} \\
B + (\epsilon_B)_j &= -\frac{\ln\left(k + (\epsilon_k)_j\right)}{\Delta t} \\
(\epsilon_B)_j &= -\left(\frac{1}{k\Delta t}\right)(\epsilon_k)_j + \mathcal{O}(\epsilon_{k,j}^2) \qquad (2.72)
\end{aligned}$$

The pattern observed in Figure 2.9 can be explained using (2.72) as a an estimate for the true value of $(\epsilon_B)_j$. For each successive trial in experiment 2.3.5, Δt is half of Δt of the last trial. Since the change of the successive values in k was relatively small, the value of $\left(\frac{1}{k\Delta t}\right)$ was about twice the previous due to the change in Δt. Therefore, to make the CI of $\left\{(\epsilon_B)_j\right\}$ more precise, the CI of $\left\{(\epsilon_k)_j\right\}$ had to be at least half the CI of $\left\{(\epsilon_k)_j\right\}$ for the previous trial.

Finally, the behaviors for small values of Δt were studied. To study the range of $\Delta t = \frac{1}{4096}$ to $\Delta t = \frac{1}{16384}$, $\Delta t = \frac{1}{2048}$ was included in figures for reference. In Figure 2.14, it can be see that the credible intervals of $\left\{(\epsilon_B)_j\right\}$ and $\left\{(\epsilon_\gamma)_j\right\}$ have two negative behaviors for $\Delta t = \frac{1}{4096}$ to $\Delta t = \frac{1}{16384}$. The size of the credible intervals grow rapidly and becomes further from the true value of B and γ as Δt gets smaller. Once again, since the behavior is more pronounced in B_j, B_j was studied. As found above, the credible intervals of $\left\{(\epsilon_k)_j\right\}$ and Δt determine the credible intervals of $\left\{(\epsilon_B)_j\right\}$. Therefore, $\left\{(\epsilon_k)_j\right\}$ and $\left\{(k)_j\right\}$ were plotted in Figure 2.15. Examining Figure 2.15, it can be seen that the precision of the credible intervals remain relatively constant while the accuracy degrades as Δt gets smaller. Then, the transformation described by (2.72) amplified these errors. It is interesting to note that the credible intervals of $\{k_j\}$ for $\Delta t = \frac{1}{4096}$ to $\Delta t = \frac{1}{16384}$ are very similar.

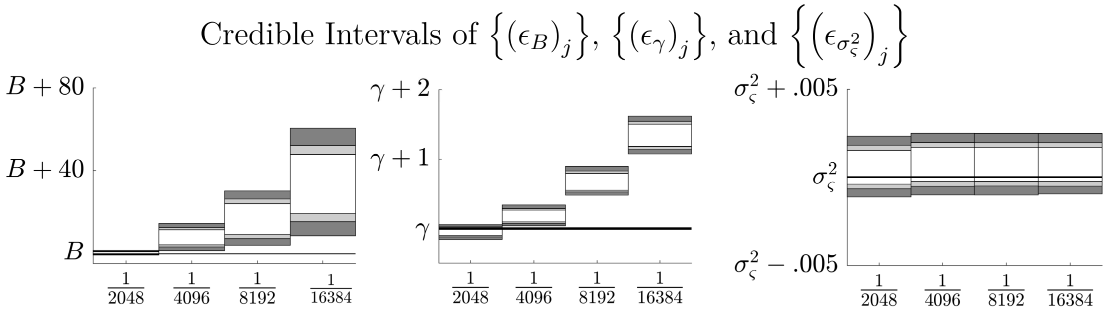
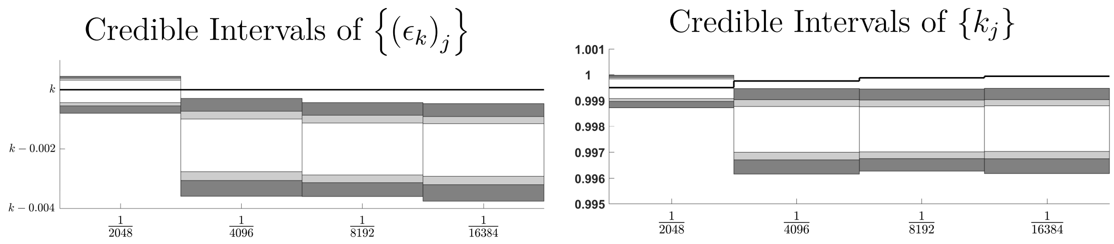


Figure 2.14: $\left\{(\epsilon_B)_j\right\}$, $\left\{(\epsilon_\gamma)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ for $\Delta t = \frac{1}{2048}$ to $\Delta t = \frac{1}{16384}$ of experiment 2.3.5

Given this odd behavior, additional tests were conducted. First, it was tested if the observed behavior was a result of the MCMC simulation and not the dataset. Following this, three more independent MCMC were run for the same dataset as above for $\Delta t = \frac{1}{4096}$ where the initial guess was the true values of the parameters. The additional MCMC simulations were plotted in Appendix B.2.3.6. Each of the independent simulations using the true values of the parameters as the initial guess produced nearly



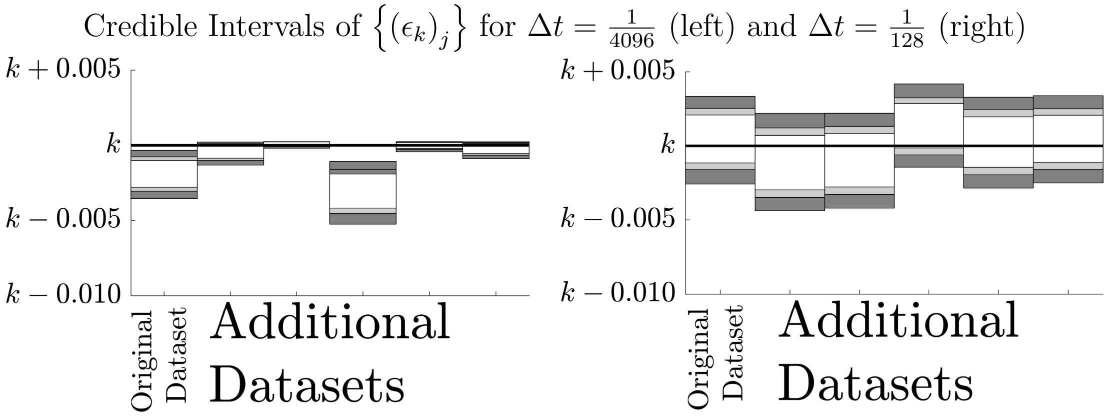
Credible Intervals of $\left\{ (\epsilon_k)_j \right\}$

Credible Intervals of $\{k_j\}$

Figure 2.15: $\left\{ (\epsilon_k)_j \right\}$ and $\left\{ (k)_j \right\}$ for $\Delta t = \frac{1}{2048}$ to $\Delta t = \frac{1}{16384}$ of experiment 2.3.5

identical results. Therefore, it was reasonable to conclude that the behavior exhibited was not a result of poorly tuned MCMC simulation.

Following the previous result, it was checked to see if other DLMs generated with 10,000 observations and $\Delta t = \frac{1}{4096}$ would show similar qualities. Since $\{B_j\}$ was the most problematic and was determined by $\{k_j\}$, the credible intervals of $\{k_j\}$ were plotted in Figure 2.16. In Figure 2.16, it was seen that only one other DLM produced results similar to the original $MCMC$ and the size of the credible intervals varied greatly.



Credible Intervals of $\left\{ (\epsilon_k)_j \right\}$ for $\Delta t = \frac{1}{4096}$ (left) and $\Delta t = \frac{1}{128}$ (right)

Figure 2.16: CI of $\left\{ (\epsilon_k)_j \right\}$ for $\Delta t = \frac{1}{4096}$ and $\Delta t = \frac{1}{128}$ comparing four new datasets to the original dataset

To check to see if this behavior was constant for all Δt, $MCMC$ simulations were run on additional datasets with $\Delta t = \frac{1}{128}$ and 10,000 observations. The credible intervals of $\left\{ (\epsilon_k)_j \right\}$ for $\Delta t = \frac{1}{128}$ were also placed in Figure 2.16. $\Delta t = \frac{1}{128}$ had credible intervals of consistent size and the true value of k_j varied randomly within the credible intervals. The poor behavior observed with the trials where $\Delta t = \frac{1}{4096}$ was caused by k being too close to 1 or Δt being too small. This behavior was problematic for two reasons. First, because without context, it would be hard to identify. The histograms of $\left\{(\epsilon_k)_j\right\}$, $\left\{\left(\epsilon_{\sigma^2_\eta}\right)_j\right\}$, and $\left\{\left(\epsilon_{\sigma^2_\varsigma}\right)_j\right\}$ for the original dataset found in Appendix B.2.3.6 do not exhibit obviously poor behavior like the cases when Δt is too large. Second, because Δt is small, this poor behavior was greatly magnified by the transformation from k to B. This was illustrated by the comparison of $g\left(\left\{(\epsilon_B)_j\right\}\right)$ for $\Delta t = \frac{1}{4096}$ and $\Delta t = \frac{1}{128}$ in Figure 2.17. Therefore, it is important to provide context by evaluating multiple datasets or choices of Δt to confirm that an appropriate value for Δt is being used.

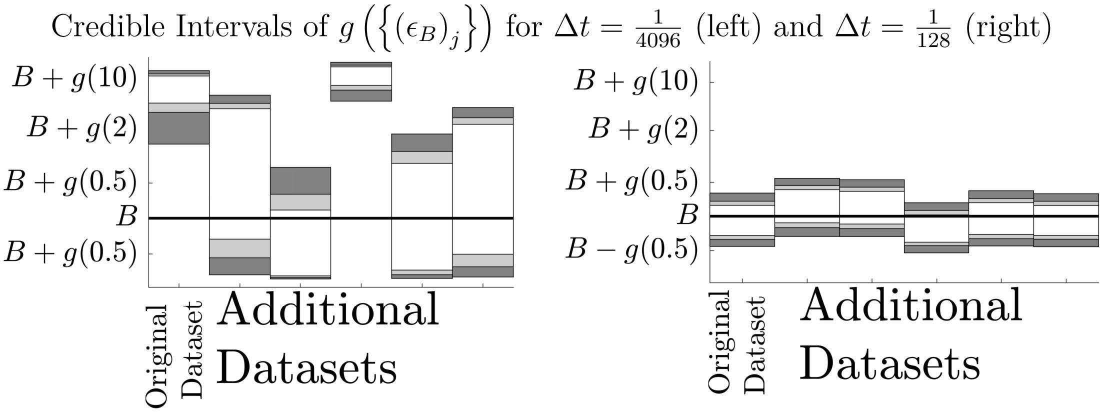


Figure 2.17: CI of $g\left(\left\{(\epsilon_B)_j\right\}\right)$ for $\Delta t = \frac{1}{4096}$ and $\Delta t = \frac{1}{128}$ comparing four new datasets to the original dataset

Varying observations from 20 to 160,000 with $\Delta = \frac{1}{8}$

For this experiment Δt was set to $\frac{1}{8}$. Then the number of observations were varied from 20 to 160,000. Each successive trial had about twice the observations then the previous. Since Δt was constant for all trials, then $k = 0.8825$, $\sigma^2_\eta = 0.1106$, and $\sigma^2_\varsigma = 0.05$ for all trials.

In experiment 2.3.5, trials with large Δt performed poorly due to identifiably issues, which should not be problematic in any of the trials in experiment 2.3.5. However, the low number of observations could be problematic. To study this, the posteriors of k, σ_η^2, and σ_ς^2 were explored for trials with a small number of observations as those are the posteriors generated by the $MCMC$ algorithm. The histograms for $\{k_j\}$, $\left\{\left(\sigma_\eta^2\right)_j\right\}$, and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for the trial with 20 observations were placed in Figure 2.18.

Histograms of the posterior draws from $\{k_j\}$, $\left\{\left(\sigma_\eta^2\right)_j\right\}$ and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ against the true values of k, σ_η^2, and σ_ς^2

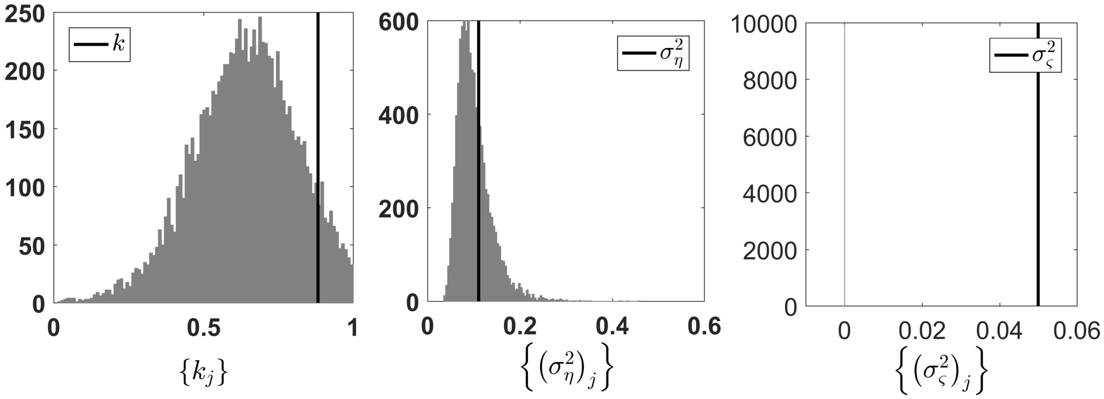


Figure 2.18: The histograms of $\{k_j\}$, $\left\{\left(\sigma_\eta^2\right)_j\right\}$, and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for 20 observations

The posterior distributions for k and σ_η^2 in Figure 2.18 had large credible intervals, but that was reasonable given there are only 20 observations. However, inference on the posterior of σ_ς^2 did not perform as well. The entire posterior of σ_ς^2 was very close to 0, and this could be caused by insufficient information within the trial. To study the sufficient number of observations for this experiment the histograms $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ were plotted for 39 to 625 in Figure 2.19.

Figure 2.19 shows that a disproportionate amount of $\left(\sigma_\varsigma^2\right)_j$ near 0 occurred when there was an insufficient number of observations. For $\Delta t = \frac{1}{8}$, poor behavior was exhibited for 20 to 312 observations. The trials with 625 observations exhibited more positive behaviors. Therefore, trials with 625 observations or more were evaluated comparing credible intervals as before. The credible intervals of $\left\{\left(\epsilon_B\right)_j\right\}$, $\left\{\left(\epsilon_\gamma\right)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ were placed in Figures 2.20, 2.21, and 2.22.

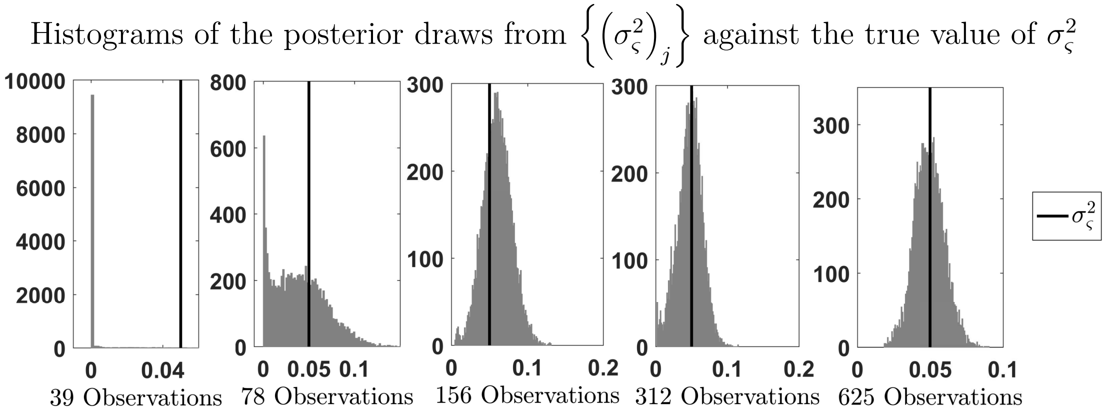


Figure 2.19: The histograms of $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for 39 to 625 observations

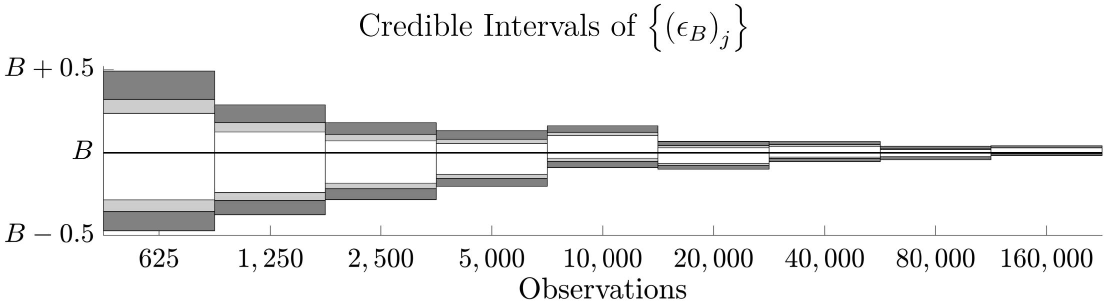


Figure 2.20: $\left\{(\epsilon_B)_j\right\}$ for 625 to 10,000 observations

In Figures 2.20, 2.21, and 2.22 the credible interval became more precise for $\left\{(\epsilon_B)_j\right\}$, $\left\{(\epsilon_\gamma)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$. Since k and Δt remain constant and $(\epsilon_B)_j \approx \left[\frac{1}{k(\Delta t)}\right](\epsilon_k)_j$, the factor of improvement of the precision of the credible interval from the discrete model was translated to the continuous variables in the stochastic differential equation. There could be a point where increasing data without changing Δt has diminishing returns, but a DLM of $1,280,000$ observations was studied in the experiment in Section 2.3.5 and it still improved over the datasets preceding. Given the computational resources available, it was not practical to search for the point where increasing data without changing Δt did not improve the precision of the credible intervals.

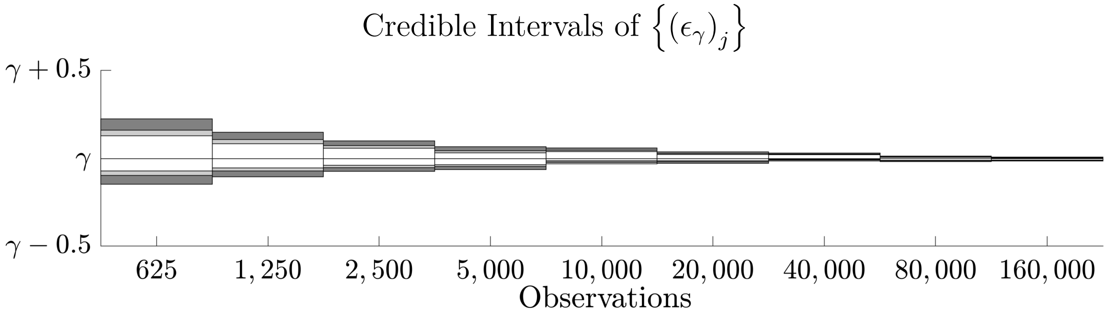


Figure 2.21: $\left\{(\epsilon_\gamma)_j\right\}$ for 625 to $10,000$ observations

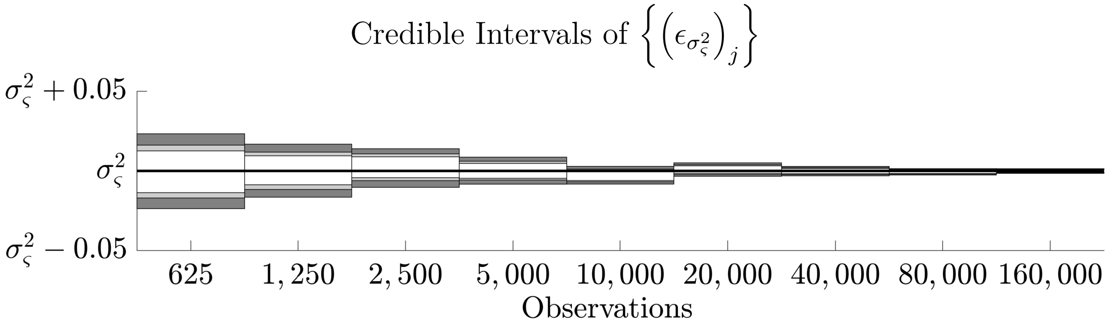
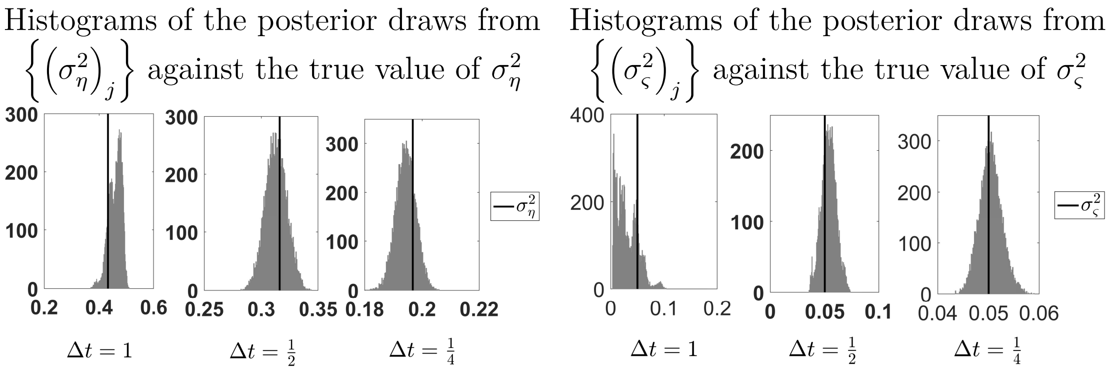


Figure 2.22: $\left\{\left(\epsilon_{\sigma^2_\varsigma}\right)_j\right\}$ for 625 to $10,000$ observations

Varying $\Delta t = 1$ to $\Delta t = \frac{1}{128}$ for a Dynamic Linear Models where $t_F - t_1 = 10,000$

The total time lapse in experiment 2.3.5 was $10,000$. The experiment with the largest Δt was 1, producing $10,000$ observations. The dataset was the same for each trial, and the trials only differed by Δt between each and the number of observations. For each trial, Δt was $\frac{1}{2}$ of the Δt of the previous trial. The smallest Δt was $\Delta t = \frac{1}{128}$ and $1,280,000$ observations were taken.

First, it was checked if identifiability issues occurred with σ^2_η and σ^2_ς for large Δt. Histograms of $\left\{\left(\sigma^2_\eta\right)_j\right\}$ and $\left\{\left(\sigma^2_\varsigma\right)_j\right\}$ were plotted in Figure 2.23 for $\Delta t = 1$ to $\Delta t = \frac{1}{4}$. The trial with $\Delta t = 1$ exhibited the aforementioned identifiability issues where the others did not.

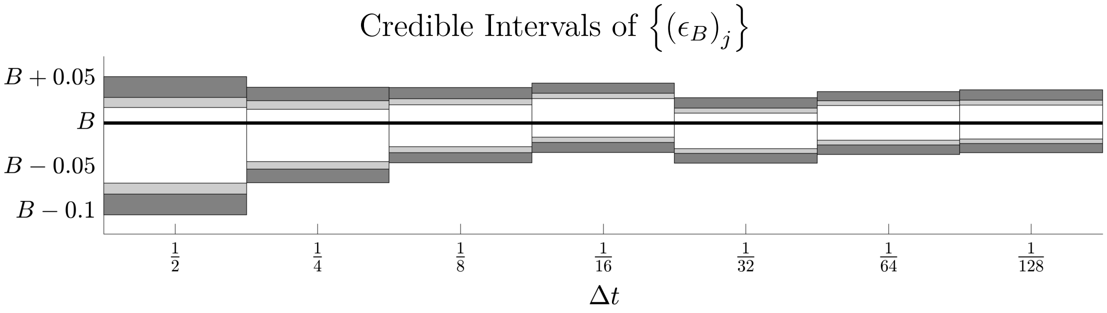


Histograms of the posterior draws from $\left\{\left(\sigma_\eta^2\right)_j\right\}$ against the true value of σ_η^2

Histograms of the posterior draws from $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ against the true value of σ_ς^2

Figure 2.23: Histograms of $\left\{\left(\sigma_\eta^2\right)_j\right\}$ and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ for $\Delta t = 1$ to $\Delta t = \frac{1}{4}$

For small Δt, the trials became to computationally expensive before the posteriors exhibited similar behavior to trials in Section 2.3.5 for small Δt. Therefore, $\Delta t = \frac{1}{2}$ to $\Delta t = \frac{1}{128}$ were evaluated using credible intervals as before. The comparison of the intervals of $\left\{(\epsilon_B)_j\right\}$, $\left\{(\epsilon_\gamma)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ were placed in Figures 2.24, 2.25, and 2.26 respectively. The credible intervals of $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ for $\Delta t = 1$ made it difficult to observe the credible intervals for all Δt. Therefore, $\left\{\left(\epsilon_{\sigma_\varsigma^2}\right)_j\right\}$ was plotted using the transformation $f(x) = g(250x)$.



Credible Intervals of $\left\{(\epsilon_B)_j\right\}$

Figure 2.24: $\left\{(\epsilon_B)_j\right\}$ for $\Delta t = \frac{1}{4}$ to $\Delta t = \frac{1}{128}$

Unlike experiment 2.3.5, the credible intervals for $\left\{(\epsilon_B)_j\right\}$ do not get less precise as Δt gets small. The credible intervals show improvement with diminishing returns as Δt gets very small. Furthermore, $\left\{(\epsilon_\gamma)_j\right\}$ gets more precise as Δt gets small. Equa-

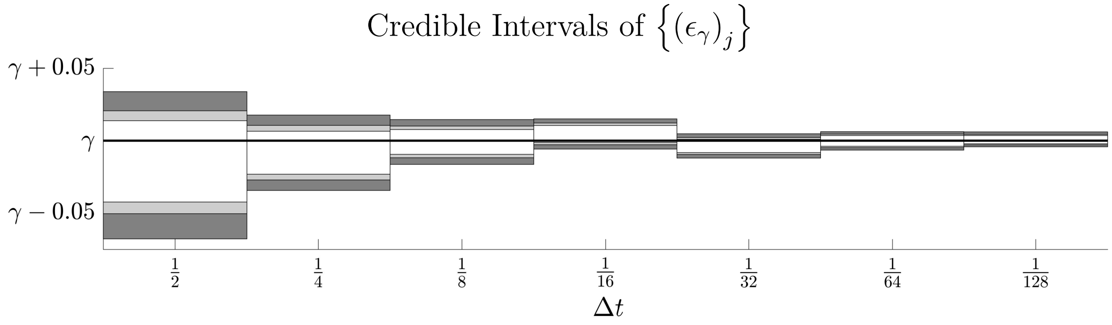


Figure 2.25: $\left\{\left(\epsilon_{\gamma}\right)_j\right\}$ for $\Delta t = \frac{1}{4}$ to $\Delta t = \frac{1}{128}$

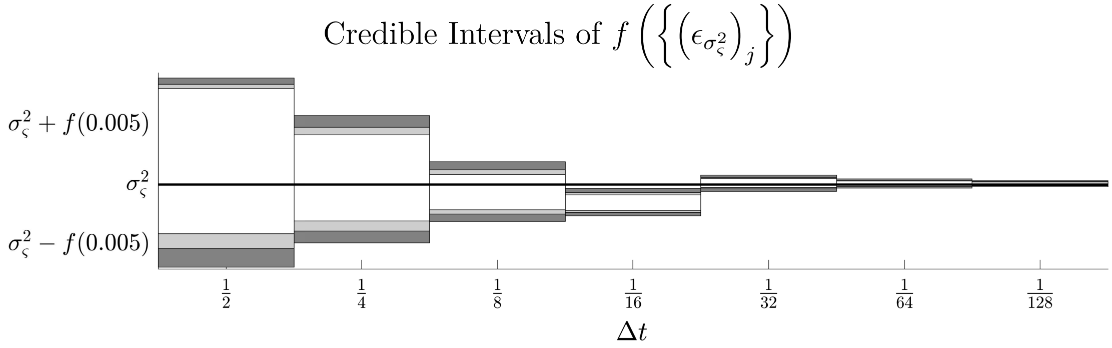


Figure 2.26: $f\left(\left\{\left(\epsilon_{\sigma_{\varsigma}^2}\right)_j\right\}\right)$ for $\Delta t = \frac{1}{4}$ to $\Delta t = \frac{1}{128}$

tion (2.72) did provide better understanding for this behavior, and this was be explored further for all experiments in the following section.

Optimizing the choice of Δt

Given the trial being examined behaved well, the mean of the $\left\{\left(\epsilon_B\right)_j\right\}$, $\left\{\left(\epsilon_{\gamma}\right)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_{\varsigma}^2}\right)_j\right\}$ should fluctuate randomly around the B, γ, and σ_{ς}^2. Since the point estimates have the aforementioned inherent randomness, the precision of the credible intervals of $\left\{\left(\epsilon_B\right)_j\right\}$, $\left\{\left(\epsilon_{\gamma}\right)_j\right\}$, and $\left\{\left(\epsilon_{\sigma_{\varsigma}^2}\right)_j\right\}$ were used to study the quality of inference for trials that did not exhibit negative behaviors. This was done by comparing $DQ\left(p; w_j\right)$ where $DQ\left(p; w_j\right)$ was defined in (2.73) and $Q\left(n; \left\{w_j\right\}\right)$ is the quantile function of the simulated distribution of $\{w_j\}$ evaluated at n.

$$DQ\left(p; w_j\right) = \frac{1}{w}\left[Q\left(1 - \frac{1-p}{2}; \{w_j\}\right) - Q\left(\frac{1-p}{2}; \{w_j\}\right)\right] \tag{2.73}$$

Figures 2.27 compares $DQ\left(.9; B_j\right)$, $DQ\left(.9; \gamma_j\right)$, and $DQ\left(.9; \left(\sigma^2_\varsigma\right)_j\right)$ versus Δt for the experiments 2.3.5 and 2.3.5.

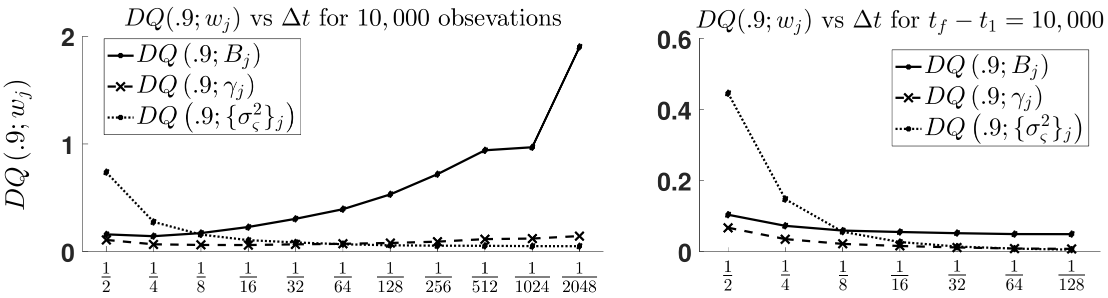


Figure 2.27: $DQ\left(.9; B_j\right)$, $DQ\left(.9; \gamma_j\right)$, and $DQ\left(.9; \left(\sigma^2_\varsigma\right)_j\right)$ versus Δt for experiments 2.3.5 (left) and 2.3.5 (right)

As discussed in previous sections and observed in Figure 2.27 the transformation from k_j to B_j was most sensitive to changes Δt for (2.70). Therefore, if the credible interval of $\{B_j\}$ was maintaining precision or becoming more precise as Δt gets small, it was reasonable to assume the same was true for γ and σ^2_ε. Following this, the credible intervals of $\{B_j\}$ were studied more extensively. Recall, that equation (2.72) approximated $\left(\epsilon_B\right)_j$ by expanding around the true value of k.

$$\left(\epsilon_B\right)_j = -\left(\frac{1}{k\Delta t}\right)\left(\epsilon_k\right)_j + \mathcal{O}\left(\left(\epsilon_k\right)_j^2\right) \tag{2.72 Revisited}$$

Following this, the value of $\frac{1}{k\Delta t}$ for each trial in the experiment from Section 2.3.5 was put in a table along side the values of $DQ\left(.9; k_j\right)$ and $DQ\left(.9; B_j\right)$.

Trial	Δt	$DQ\left(.9;k_j\right)$	$\frac{1}{k\Delta t}$	$DQ\left(.9;B_j\right)$
1	$\frac{1}{2}$	0.048081	3.2976	0.159657
2	$\frac{1}{4}$	0.027837	5.1361	0.142584
3	$\frac{1}{8}$	0.018837	9.0652	0.171421
4	$\frac{1}{16}$	0.013332	17.0321	0.228052
5	$\frac{1}{32}$	0.009198	33.0169	0.304173
6	$\frac{1}{64}$	0.006054	65.0076	0.393446
7	$\frac{1}{128}$	0.004117	129.0062	0.530904
8	$\frac{1}{256}$	0.002801	257.0023	0.719547
9	$\frac{1}{512}$	0.001835	513.0261	0.940986
10	$\frac{1}{1024}$	0.000946	1,025.025	0.969292
11	$\frac{1}{2048}$	0.000928	2,049.025	1.901532

Table 2.1: Comparing the ranges of $DQ\left(.9;k_j\right)$, $\frac{1}{k\Delta t}$, $DQ\left(.9;B_j\right)$ for the experiments from 2.3.5

In Table 2.1, it was seen that as Δt get small, $\frac{1}{k\Delta t}$ was about twice the previous for each successive trial. Therefore, to become more precise for small Δt, the $DQ\left(.9;k_j\right)$ had to be less than half of the previous trial. $DQ\left(.9;k_j\right)$ was highly dependent on the noise of x_i $\left(\sigma_\eta^2\right)$. σ_η^2 decreases for each successive trial of experiment 2.3.5 which in turn resulted in more precise credible intervals of k_j. However, the number of observations for each trial is $10,000$. Each successive trial observed the continuous behavior for half the time. Therefore, for improvement, the reduced value of σ_η^2 had to outweigh the adverse effect of observing the system for less time. Following this, although the credible intervals of k for experiment 2.3.5 in Table 2.1 became more precise; it was not by a large enough factor that the credible intervals of B_j became more precise. For the given parameters and observations, the credible interval of B_j did not become more precise after $\Delta t = \frac{1}{4}$.

In the experiment from Section 2.3.5, $t_f - t_0 = 10,000$ for each trial. Therefore, each successive trial had twice as many observations. Unlike the previous experiment, in each successive trial, σ_η^2 was decreasing, and there was the same amount of time observed. As observed in Table 2.2, $DQ\,(.9; k_j)$ for each successive trial was about half or less than half of the previous trial so the credible interval of B_j was more precise. Furthermore, in Figure 2.27 it was seen that $\{\gamma_j\}$ and $\left\{\left(\sigma_\varsigma^2\right)_j\right\}$ became more precise for each successive trial in experiment 2.3.5.

Trial	Δt	$DQ\,(.9; k_j)$	$\frac{1}{k\Delta t}$	$DQ\,(.9; B_j)$
1	$\frac{1}{2}$	0.031678	3.2976	0.103580
2	$\frac{1}{4}$	0.014133	5.1361	0.072369
3	$\frac{1}{8}$	0.006518	9.0652	0.059068
4	$\frac{1}{16}$	0.003229	17.0321	0.055025
5	$\frac{1}{32}$	0.001565	33.0169	0.051642
6	$\frac{1}{64}$	0.000756	65.0076	0.049150
7	$\frac{1}{128}$	0.000379	129.0062	0.048938

Table 2.2: Comparing the ranges of $DQ\,(.9; k_j)$, $\frac{1}{k\Delta t}$, $DQ\,(.9; B_j)$ for the experiments from 2.3.5

In Table 2.2, it can be seen that there are diminishing returns for B. As Δt gets smaller, the precision of of B_j improves by less (although γ_j and $\left(\sigma_\varsigma^2\right)_j$ showed greater improvement). Unfortunately, this added precision is not free. As stated above, each successive trial has more observations. In this experiment, computation time became problematic before other factors. Figure 2.28 shows the computation time in hours from experiment 2.3.5.

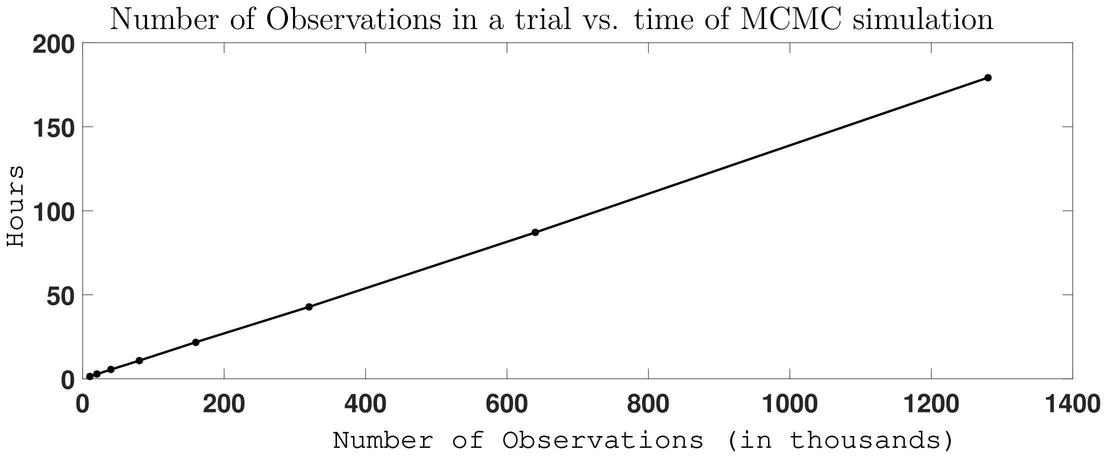


Figure 2.28: A comparison of number of observations and computation time.

Finally, the experiment in Section 2.3.5 did not require advanced analysis. For that experiment, more data was added with no cost. Therefore, the only limiting factors were simply computation time and data available.

For the parameters $B = \gamma = 1$, B was the most difficult to make inference on. Therefore, B was used as a baseline for decisions on Δt and $t_F - t_1$ while factoring in computation times. Finally, for the case of small Δt, as occurred in the experiment in Section 2.3.5, poor behavior could be difficult to identify. Therefore, it was important to add enough context for that case to rule out those negative behaviors.

2.4 Inference on a discrete state space continuous time Markov given observations in discrete time

2.4.1 Continuous time Markov chains with k discrete states and k signals

The last section introduced the basics of using a HMM (hidden Markov model) to make inference on a continuous state, continuous time system with measurement noise. However, the applications which were the focus of this research had discrete state spaces. The two models of interest were the continuous time systems with two or three discrete states as diagrammed in Figures 2.29A and 2.29B respectively. Although the parameters of interest were the parameters governing the continuous time system, inference was not made on the continuous time model directly. Like the DLM, inference was made on the time discretization of that model. This section discusses the mapping between the continuous time model and the discrete time model.

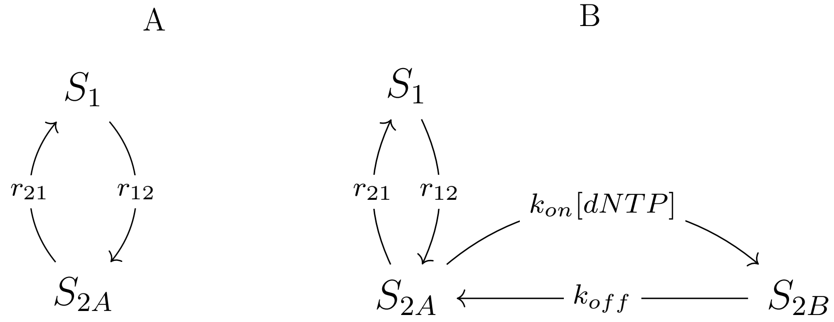


Figure 2.29: Diagrams of the two state model (A) and three state model (B) (Revisited)

As discussed in Section 1.1, the three state diagram in Figure 2.29B only had two distinct emissions. However, for the purpose of introducing this topic, only systems that had a distinct emission for each state were discussed.

Transition probabilities of Markov models given a two state system

First, the two state model diagrammed in Figure 2.29A was explored. In Figure 2.29A, r_{12} and r_{21} represented exponential transition rates and $X(t) = 1$ denotes the system is in state S_1 and $X(t) = 2$ denotes the system is in state S_{2A} at time t. Since the transition rates were exponential, the two state model was Markovian or $p(X(t_{i+2})|X(t_{i+1}), X(t_i)) = p(X(t_{i+2})|X(t_{i+1}))$ where $t_i \leq t_{i+1} \leq t_{i+2}$. The exponential transition rates of the system diagrammed in 2.29A were equivalent to a constant change in probability. Therefore, the system could be expressed efficiently as a linear system of differential equations as in (2.74) and (2.75). For brevity, $P(t)_j = P(X(t) = j)$.

$$\frac{dP(t)_1}{dt} = -r_{12}P(t)_1 + r_{21}P(t)_2 \tag{2.74}$$

$$\frac{dP(t)_2}{dt} = r_{12}P(t)_1 - r_{21}P(t)_2 \tag{2.75}$$

Unfortunately, the model was not measured continuously. As with the DLM, the state was measured at $\boldsymbol{t} = (t_1, t_2, ..., t_T)$ where $t_{i+1} - t_i = \Delta t$ for all $i \in \{1, 2, 3, ..., T-1\}$. Therefore, it was necessary to describe (2.74) and (2.75) at each t_i or in discrete time. Given the exponential dwell times and regularly collected data, the discrete description was also Markovian. Therefore, the transition probabilities from state m to state n are equal for all t_i or $p(X(t_i) = m|X(t_{i-1}) = n) = p(X(t_l) = m|X(t_{l-1}) = n)$ for all i,l $\in \{2, 3, ..., T\}$ and $m, n \in \{1, 2\}$. Let q_{12} denote the probability of transition from state S_1 to S_{2A} and q_{21} denote the probability of transition from state S_{2A} to S_1. Using this notation, q_{12} and q_{21} were expressed in terms of the continuous system and r_{12} and r_{21} were expressed in terms of the discrete system. These were listed as (2.76), (2.77), (2.79), and (2.80) which were derived by solving (2.74) and (2.75). The derivation can be found in Appendix C.1.

$$
\begin{aligned}
q_{12} &= \frac{r_{12}}{r_{12}+r_{21}}\left(1-\exp(-\Delta t(r_{12}+r_{21}))\right) && (2.76)\\
q_{21} &= \frac{r_{21}}{r_{12}+r_{21}}\left(1-\exp(-\Delta t(r_{12}+r_{21}))\right) && (2.77)\\
r_{12}+r_{21} &= -\frac{\ln(1-[q_{12}+q_{21}])}{\Delta t} && (2.78)\\
r_{12} &= \frac{q_{12}(r_{12}+r_{21})}{1-\exp(-\Delta t(r_{12}+r_{21}))} && (2.79)\\
r_{21} &= \frac{q_{21}(r_{12}+r_{21})}{1-\exp(-\Delta t(r_{12}+r_{21}))} && (2.80)
\end{aligned}
$$

The likelihood of the two discrete state, discrete time system

Given that the two state discrete system is memoryless and the transition probabilities from state m to n are q_{mn}, the likelihood could be written. For this, let $\boldsymbol{z_i} = (z_{i1}, z_{i2})'$ be the discrete state measurement where $\boldsymbol{z}_i = (1,0)'$ if $X(t_i) = 1$ and $\boldsymbol{z}_i = (0,1)'$ if $X(t_i) = 2$. So the model generalizes easily to k states, let $z_{im} = 1$ denote that $z_{im} = 1$ and $z_{in} = 0$ for all $n \neq m$. Using this notation the probability of the trace of the states given T observations and $\boldsymbol{\theta}$ were $p(\boldsymbol{z}_1, \boldsymbol{z}_2, ...\boldsymbol{z}_T|\boldsymbol{\theta}) = p(\boldsymbol{z}_1, \boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}, \boldsymbol{\theta})$. From this $p(\boldsymbol{z}_1, \boldsymbol{z}_2, ...\boldsymbol{z}_T)$ was defined in (2.81) where $\boldsymbol{z}_1^T = (\boldsymbol{z}_1, \boldsymbol{z}_2, ...\boldsymbol{z}_T)$.

$p(\boldsymbol{z}_1^T|\boldsymbol{\theta})$ for 2 states (2.81)

$$
\begin{aligned}
p(\boldsymbol{z}_1^T|\boldsymbol{\theta}) &= p(\boldsymbol{z}_1, \boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1})\\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) &= p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)1} = 1)^{\boldsymbol{z}_{(i-1)1}} p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)2} = 1)^{\boldsymbol{z}_{(i-1)2}}\\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)1} = 1) &= (1-q_{12})^{z_{i1}}(q_{12})^{z_{i2}}\\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)2} = 1) &= (q_{21})^{z_{i1}}(1-q_{21})^{z_{i2}}\\
p(\boldsymbol{z}_1|\boldsymbol{\theta}) &= \rho_1^{z_{11}}\rho_1^{z_{12}}
\end{aligned}
$$

$p(\boldsymbol{z}_1|\boldsymbol{\theta})$ had to be treated differently than the $\boldsymbol{z}_2^T$. That was because there was no state prior to $\boldsymbol{z}_1$. Therefore, the associated transition probabilities from the state prior to $\boldsymbol{z}_1$ did not exist. Following this, a parameter $\boldsymbol{\rho}$ was used where ρ_m was the probability

$z_{1m} = 1$.

As discussed in Section 1.1, the state was not directly observed. Instead, each state m at time i produced a signal (y_i) that was normally distributed with mean μ_m and variance σ^2. Therefore, it was necessary to consider the joint probability of $\boldsymbol{z}_1^T$ and $\boldsymbol{y}_1^T$ which was listed in (2.82). The undefined probability distributions in (2.82) were as defined in (2.81).

$$p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T | \boldsymbol{\theta}) \text{ for 2 states } \quad (2.82)$$

$$\begin{aligned}
p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T | \boldsymbol{\theta}) &= p(\boldsymbol{z}_1, \boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i | \boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i | \boldsymbol{z}_i, \boldsymbol{\theta}) \\
y_i | \boldsymbol{z}_{im} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_m, \sigma^2) \\
p(y_i | \boldsymbol{z}_i, \boldsymbol{\theta}) &= \left[p(y_i | \boldsymbol{z}_{i1} = 1, \boldsymbol{\theta}) \right]^{z_{i1}} \left[p(y_i | \boldsymbol{z}_{i2} = 2, \boldsymbol{\theta}) \right]^{z_{i2}}
\end{aligned}$$

Transition probabilities of Markov models given a k state system

Although this research only explored two and three state systems, the algorithms employed generalize easily to k states. Therefore, inference on the discrete k state, continuous time system diagrammed in Figure 2.30 was explored. Like the two state system, the discrete k state system was determined by exponential transition rates between adjacent states. Following this, the discrete k state, continuous time system was also Markovian. This type of system that was both Markovian and measured on the continuous time scale is commonly known as a continuous time Markov chain (CMTC).

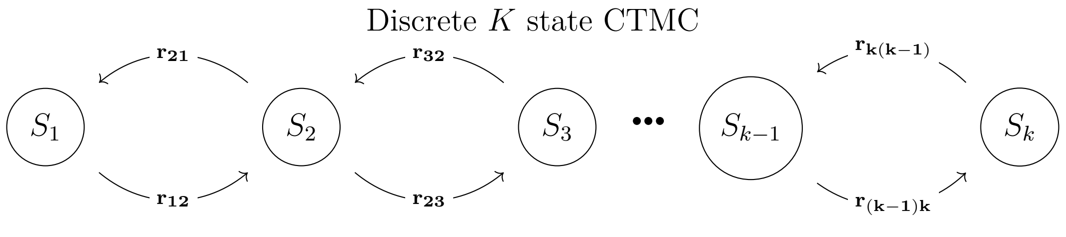


Figure 2.30: A diagram of a system with k discrete states that produce k distinct signals

As with the two state system, the k state CTMC can be expressed as a linear system of differential equations which were listed as (2.83).

System of differential equations representation of the discrete k state CTMC (2.83)

$$\begin{aligned}
\frac{dP_1(t)}{dt} &= -r_{12}P(t)_1 + r_{21}P(t)_2 \\
\frac{dP_2(t)}{dt} &= r_{12}P_1(t) - (r_{21} + r_{23})P_2(t) + r_{32}P_3(t) \\
\vdots \qquad & \qquad \vdots \\
\frac{dP_j(t)}{dt} &= r_{(j-1)j}P_{j-1}(t) - (r_{j(j-1)} + r_{j(j+1)})P_j(t) + r_{(j+1)j}P_j(t) \\
\vdots \qquad & \qquad \vdots \\
\frac{dP_k(t)}{dt} &= r_{(k-1)k}P_{k-1}(t) - r_{k(k-1)}P_k(t)
\end{aligned}$$

In the case of k discrete states, it was more efficient to write (2.83) as $\frac{d\mathbf{P}(t)}{dt} = \mathbf{R}\mathbf{P}(t)$ where $\mathbf{P}(t)$ was defined in (2.84) and $\mathbf{R}$ was a tridiagonal matrix defined in (2.85).

$$\mathbf{P}(t) = \begin{bmatrix} P_1(t) & P_2(t) & \cdots & P_k(t) \end{bmatrix}' \tag{2.84}$$

$$\boldsymbol{R} = \begin{bmatrix}
-r_{12} & r_{21} & & & \\
r_{12} & -(r_{21} + r_{23}) & r_{32} & & \\
 & r_{23} & -(r_{32} + r_{34}) & r_{43} & \\
 & & \ddots & \ddots & \ddots \\
 & & r_{(k-2)(k-1)} & -(r_{(k-1)(k-2)} + r_{(k-1)k}) & r_{k(k-1)} \\
 & & & r_{(k-1)k} & -r_{k(k-1)}
\end{bmatrix} \tag{2.85}$$

The exact solution to $\frac{d\mathbf{P}(t)}{dt} = \mathbf{R}\boldsymbol{P}(t)$ is known to be $\boldsymbol{P}(t_i) = e^{\mathbf{R}(t_i - t_{i-1})}\mathbf{P}(t_{i-1})$ over the interval $t_i - t_{i-1} = \Delta t$ where $e^{\boldsymbol{R}\Delta t}$ is the matrix exponential and $\boldsymbol{P}(t_{i-1})$ contains the initial conditions. Given that the system is Markovian, this was written equivalently as $\boldsymbol{P}(\Delta t) = e^{\mathbf{R}\Delta t}\mathbf{P}(0)$ where $\boldsymbol{P}(0)$ is the initial condition at any time t_{i-1} and $\boldsymbol{P}(\Delta t)$ contains the probabilities of being in state n at t_i.

A discrete time Markov chain with measurement of state at all t_i and time interval Δt can be described by the equation $\boldsymbol{P}(\Delta t) = \boldsymbol{Q}(\Delta t)\mathbf{P}(0)$ where $\boldsymbol{Q}(\Delta t)$ is known as the transition probability matrix. The entry in the n^{th} row and the m^{th} column of the matrix is the transition probability from state m to n which was denoted as q_{mn}. Given that $\boldsymbol{P}(\Delta t) = \boldsymbol{Q}(\Delta t)\mathbf{P}(0)$ and $\boldsymbol{P}(\Delta t) = e^{\mathbf{R}\Delta t}\mathbf{P}(0)$, $\boldsymbol{Q}(\Delta t) = e^{\mathbf{R}\Delta t}$. Therefore, to compute $\boldsymbol{Q}(\Delta t)$ it was only necessary to compute $e^{\mathbf{R}\Delta t}$.

In the case of the matrix $\boldsymbol{R}$, the analytic form of $e^{\mathbf{R}\Delta t}$ was not straightforward. Since the matrix exponential is defined as $\sum_{d=0}^{\infty} \frac{(\mathbf{R}(\Delta t))^d}{d!}$, then $\boldsymbol{Q}(\Delta t)$ can be approximated by the first d terms of the series if Δt is sufficiently small.

$$\boldsymbol{Q}(\Delta t) = \sum_{d=0}^{D} \frac{(\mathbf{R}(\Delta t))^d}{d!} + o(\Delta t^D) \tag{2.86}$$

Given that inference was made on $\boldsymbol{Q}(\Delta t)$, it was also necessary to be able to solve for $\boldsymbol{R}$ given $\boldsymbol{Q}(\Delta t)$. The transition rates in $\boldsymbol{R}$ were independent of Δt and were approximated from (2.86). Using (2.86), $\boldsymbol{R}$ was approximated using an iterative process with error $o(\Delta t^D)$ where $D+1$ was the number of iterations computed.

$$\begin{aligned}
\boldsymbol{R} &= \boldsymbol{R}^{(3)} + o\left(\Delta t^3\right) \qquad (2.87)\\
\boldsymbol{R}^{(3)} &= \boldsymbol{R}^{(0)} - \frac{1}{2}(\Delta t)\left(\boldsymbol{R}^{(0)}\right)^2 + \frac{1}{3}(\Delta t)^2\left(\boldsymbol{R}^{(0)}\right)^3 - \frac{1}{4}(\Delta t)^3\left(\boldsymbol{R}^{(0)}\right)^4\\
\boldsymbol{R}^{(0)} &= \frac{\boldsymbol{Q}-\boldsymbol{I}}{\Delta t}
\end{aligned}$$

(2.87) was derived in Appendix C.2. For the work in this research, Δt was set to $\frac{1}{100{,}000}$. Therefore, by including four terms for both (2.86) and (2.87) the error introduced by approximation was negligible relative to the uncertainty introduced in the inference.

The likelihood of the k discrete state, discrete time system

Following the two state system, the k discrete state, discrete time system was also a Markov chain. Furthermore, the notation scales to k states and $\boldsymbol{z}_i$ carried the same meaning as previously where $\boldsymbol{z}_i = (z_{i1}, ...z_{ik})$. It was important to note, the tridiagonal properties of $\boldsymbol{R}$ were not upheld by $\boldsymbol{Q}$. That was because it was possible to have multiple state changes occur in one Δt. Therefore, all q_{mn} were nonzero and should be computed if possible.

Aside from that difference, the likelihood was identical to the two discrete state, discrete time system. The likelihood for k states was listed as (2.88).

$$p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta}) \text{ for } k \text{ states} \quad (2.88)$$

$$\begin{aligned}
p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta}) &= p(\boldsymbol{z}_1|\boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) &= \prod_{j=1}^{k} \Big[p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)j} = 1)\Big]^{\boldsymbol{z}_{(i-1)j}} \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)j} = 1) &= \prod_{l=1}^{k} q_{jl}^{z_{il}} \\
p(\boldsymbol{z}_1|\boldsymbol{\theta}) &= \prod_{j=1}^{k} \rho_j^{z_{1j}} \\
y_i|\boldsymbol{z}_{ij} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_j, \sigma^2) \\
p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) &= \prod_{j=1}^{k} [p(y_i|z_{ij} = 1, \boldsymbol{\theta})]^{z_{ij}}
\end{aligned}$$

As with the DLM, the k discrete state, discrete time system was expressed as a joint probability where $\boldsymbol{z}_1^T$ was latent. Therefore, inference was made on both $\boldsymbol{z}_1^T$ and $\boldsymbol{\theta}$ iteratively. For this, a special case of the EM algorithm called the Baum-Welch algorithm, a similar technique to the EM method called Viterbi training, and a version of the forward-filter backward sample were explored in the following sections.

2.4.2 Inference on discrete time Markov chains with k discrete states and k signals

As with the DLM, the goal was to make inference on the transition probabilities and signals. Given the joint probability with a latent variable, it was necessary to make inference on the latent variable as well as the parameters of interest. This was done three different ways. The first is called the Baum-Welch algorithm [4]. Although it was developed before the general EM algorithm [1], the Baum-Welch algorithm is a special case of the EM algorithm. The second method was also frequentist in nature. It works in a very similar fashion to the EM method except the most likely path of $\boldsymbol{z}_1^T$ was used instead of the expectation of $\boldsymbol{z}_1^T$. The algorithm for the most likely path is known as the Viterbi Algorithm [58] and using that path for inference on parameters has been referred to as Viterbi Training, Viterbi Extraction, or Segmental K-Means [24] [44]. Finally, the third method is fully Bayesian and employs a gibbs sampler [48] that contains a forward filter backward sampler [9].

The Baum-Welch algorithm for k states

Since the Baum-Welch algorithm is a special case of of the EM algorithm, it consists of an expectation step and a maximization step. As with all forms of the EM method, the Baum-Welch algorithm iterates between computing the E-step and the M-step.

$$\begin{aligned} &\text{EM steps for Baum-Welch Algorithm} \quad (2.89) \\ &\text{E-Step: Compute } \mathbb{E}_{\boldsymbol{z}_1^T|\boldsymbol{y}_1^T,\hat{\boldsymbol{\theta}}_{j-1}}[p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})] \\ &\text{M-Step: Compute } \hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{z}_1^T|\boldsymbol{y}_1^T,\hat{\boldsymbol{\theta}}_{j-1}}[p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})] \end{aligned}$$

The E-step consists of computing the expectation of all functions of $\boldsymbol{z}$ the joint likelihood listed as (2.88). For this, it was necessary to compute $\mathbb{E}\left[z_{im}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right]$ and $\mathbb{E}\left[z_{im}z_{(i-1)n}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right]$ for all i, m, and n.

The E-step or $\mathbb{E}\left[z_{im}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right]$ and $\mathbb{E}\left[z_{im}z_{(i-1)n}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right]$ were computed iteratively using a version of the forward backward method. In the case of the DLM, the state space was a continuous variable that was normally distributed so the expectations and variance were a convenient way to describe the state at time i. This was not the case when the state space was discrete, but the structure of $\boldsymbol{z}_i$ allowed a simple solution. Since the variable z_{im} was an indicator variable as to whether the system was is in state m at time i, $\mathbb{E}\left[z_{im}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right] = p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1})$ and $\mathbb{E}\left[z_{im}z_{(i+1)n}|\boldsymbol{y}_1^T, \hat{\boldsymbol{\theta}}_{j-1}\right] = p(z_{im} = 1, z_{(i+1)n} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1})$ where $z_{im} = 1$ denotes $z_{im} = 1$ and $z_{iv} = 0$ for all $v \neq m$. Therefore, rather than using the notation of expectations as with the DLM, the discrete space system was described in terms of probability density functions.

$$a_i(m) = p(z_{im} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta}) \tag{2.90}$$

$$\alpha_i(m) = p(z_{im} = 1, \boldsymbol{y}_1^{i}|\boldsymbol{\theta}) \tag{2.91}$$

The state update denoted as (2.90) and the state forecast denoted as (2.91) which differed from the values computed for the Kalman filter were computed from $i = 1$ to $i = T$. The algorithm for computing $a_i(m)$ and $\alpha_i(m)$ was listed as (2.92). Like the Kalman filter, the forward backward presented here can also be used to compute the probability of being in a particular state at time i when $\boldsymbol{\theta}$ was known. To maintain generality, $\boldsymbol{\theta}$ was used. However, in the case of using the entire Baum-Welch algorithm for parameter estimation, all $\boldsymbol{\theta}$ in this section should be replaced with $\hat{\boldsymbol{\theta}}_{j-1}$. A derivation of (2.92) can be found in Appendix C.3.1.

The forward iterations for the the Baum-Welch algorithm (2.92)

```
I For $m = 1$ to $k$
    (a) $a_1(m) = \rho_m$
    (b) $\alpha_1(m) = a_1(m)p(y_1|z_{im} = 1, \boldsymbol{\theta})$ where $y_1|z_{im} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_m, \sigma^2)$
II end
III For $i = 1$ to $T$
    (a) For $n = 1$ to $k$
        i. $a_i(n) = \sum_{m=1}^{k} \alpha_{i-1}(m)q_{mn}Z$
        ii. $\alpha_i(n) = a_i(n)p(y_i|z_{in} = 1, \boldsymbol{\theta})$ where $y_i|z_{in} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_n, \sigma^2)$
    (b) end
IV end
```

The backward algorithm is slightly more involved than that from the Kalman filter. To compute the necessary expectation, four different calculations were made. The definitions of three of the components were listed in (2.93)-(2.95).

$$\Gamma := p(\boldsymbol{y}_1^T|\boldsymbol{\theta}) \tag{2.93}$$

$$\gamma_i(m) = p(\boldsymbol{z}_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}) \tag{2.94}$$

$$\xi_i(m, n) = p(z_{(i+1)n} = 1, z_{im} = 1|\boldsymbol{\theta}) \tag{2.95}$$

The last component $\beta_i(m)$ did not have a convenient definition as with (2.93)-

(2.95). The most common explanations used are that $p(\boldsymbol{z}_{im} = 1, \boldsymbol{y}_1^T|\boldsymbol{\theta}) = \beta_i(m) \cdots \times \alpha_i(m)$ or $\beta_i(m) = p(\boldsymbol{y}_{i+1}^T|\boldsymbol{z}_{im} = 1, \boldsymbol{\theta})$. Although neither definition adds much intuitive understanding of the data, $\beta_i(m)$ does allow for convenient computation of the required values $\gamma_i(m)$ and $\xi_i(m,n)$. Finally, as with the Kalman filter, the last time-step was dealt with differently. That is because $\alpha_T(m) = p(\boldsymbol{z}_{Tm} = 1, \boldsymbol{y}_1^T|\boldsymbol{\theta})$. Therefore, $\beta_T(m)$ was set to one for all m. All calculations necessary for the backward algorithm were listed as (2.96). A derivation of (2.96) can be found in Appendix C.3.2.

The backwards iterations for the the Baum-Welch algorithm (2.96)

```
I $\Gamma = \sum_{m=1}^{k} \alpha_T(m)$
II For $m = 1$ to $k$
    (a) $\beta_T(m) = 1$
    (b) $\gamma_T(m) = \frac{\alpha_T(m)\beta_T(m)}{\Gamma}$
III end
IV For $i = T - 1$ to $1$
    (a) For $m = 1$ to $k$
        i. $\beta_i(m) = \sum_{m=1}^{k} q_{mn} p(y_{i+1}|z_{in} = 1, \boldsymbol{\theta})\beta_{i+1}(n)$
           where $y_{i+1}|z_{in} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_n, \sigma^2)$
        ii. $\gamma_i(m) = \frac{\alpha_i(m)\beta_i(m)}{\Gamma}$
        iii. for $n = 1$ to $k$
            A. $\xi_i(m,n) = \frac{\alpha_i(m) q_{mn} p(y_{i+1}|z_{i+1n}=1,\boldsymbol{\theta})\beta_{i+1}(n)}{\Gamma}$
               where $y_{i+1}|z_{(i+1)n} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_n, \sigma^2)$
        iv. end
    (b) end
V end
```

It is important to note that the probabilities computed in (2.96) are often very close to zero. To prevent underflow, it necessary to compute the logarithm of the probabilities in (2.92) and (2.96).

The M-step maximizes $\boldsymbol{\theta}$ for $\mathbb{E}_{\boldsymbol{z}_1^T|\boldsymbol{y}_1^T,\hat{\boldsymbol{\theta}}_{j-1}}[p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})]$. This step computes $\boldsymbol{\theta}_j$ given $\boldsymbol{\theta}_{j-1}$. Therefore, (2.93), (2.94), and (2.95) should be conditioned on $\boldsymbol{\theta}_{j-1}$ instead of $\boldsymbol{\theta}$ if estimation of $\boldsymbol{\theta}$ was the goal. The MLE of each parameter given was listed in (2.97) to (2.100) . A derivation can be found in Appendix C.3.3.

$$(\hat{\mu}_m)_j = \frac{\sum_{i=1}^T \gamma_i(m) y_i}{\sum_{i=1}^T \gamma_i(m)} \tag{2.97}$$

$$\hat{\sigma^2}_j = \frac{\sum_{i=1}^T \left[\sum_{m=1}^k \gamma_i(m)(y_i - (\hat{\mu}_m)_j)^2\right]}{T} \tag{2.98}$$

$$(\hat{q}_{mn})_j = \frac{\sum_{i=1}^{T-1} \xi_i(m,n)}{\sum_{i=1}^{T-1} \gamma_i(m)} \tag{2.99}$$

$$(\hat{\rho}_m)_j = \gamma_1(m) \tag{2.100}$$

The derivations in Appendix C.3 for the Baum-Welch algorithm follow the framework suggested by [6]. [6], [34], [4], contain more information on this type of HMM and an alternate proof can be found [34] and [4]. The entire Algorithm was listed as (2.101) where δ was an arbitrary small number.

The Baum-Welch Algorithm for k discrete states with k distinct Gaussian Emissions (2.101)

```
I Pick $\hat{\boldsymbol{\theta}}_0$
II Set $j = 0$ and define $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$.
III While $\left|\left|\hat{\boldsymbol{\theta}}_j - \hat{\boldsymbol{\theta}}_{j-1}\right|\right|_\infty > \delta$
    (a) $j = j + 1$
    (b) Calculate $\alpha_i(m)$ and $a_i(m)$ using (2.92).
    (c) Calculate $\beta_i(m)$, $\gamma_i(m)$, and $\xi_i(m,n)$ using (2.96).
    (d) for $m = 1$ to $k$
          i. Calculate $\left(\hat{\mu}_m\right)_j$ using (2.97).
         ii. Calculate $\hat{\sigma^2}_j$ using (2.98).
        iii. for $n = 1$ to $k$
               A. Calculate $\left(\hat{q}_{mn}\right)_j$ using (2.99).
         iv. end
          v. Calculate $\left(\hat{\rho}_m\right)_j$ using (2.100).
    (e) end
IV end
```

Viterbi training for k states

An alternate to the Baum-Welch algorithm can be used for computing the parameters of the k state, k distinct emission hidden Markov model. This method has been referred to as Viterbi extraction, Viterbi training, and Segmental K-means and is based on making inference on the most likely state sequence of $\boldsymbol{z}_1^T$ given the parameters. The most likely path (Viterbi path) was found employing the Viterbi algorithm [58]. This Viterbi path which was denoted as $v\left(\boldsymbol{z}_1^T\right)$ was then used to estimate the parameters in an iterative method [24] [44]. Like the EM-method, Viterbi training iteratively calculates the Viterbi path and then maximizes the parameters given the Viterbi path until convergence. An outline of the algorithm was listed as (2.102). Viterbi Training is known to be less computationally intensive than Baum-Welch, but Viterbi Training is not asymptotically unbiased [42]. Choosing between the Baum-Welch and Viterbi Training will be disused in a later section in the context of the experiments discussed in Section 1.1.

The Viterbi Training Algorithm for k discrete states with k distinct Gaussian Emissions (2.102)

```
I Pick $\hat{\boldsymbol{\theta}}_0$
II Set $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$ and $j = 0$.
III While $\left|\left|\hat{\boldsymbol{\theta}}_j - \hat{\boldsymbol{\theta}}_{j-1}\right|\right|_\infty > \delta$
    (a) $j = j + 1$
    (b) for $m = 1$ to $k$
        i. Calculate $v\left(\boldsymbol{z}_1^T\right)$ using (2.104).
        ii. Calculate $\left(\hat{\mu}_m\right)_j$ using (2.108).
        iii. Calculate $\widehat{\sigma^2}_j$ using (2.109).
        iv. for $n = 1$ to $k$
            A. Calculate $\left(\hat{q}_{mn}\right)_j$ using (2.107).
        v. end
        vi. Calculate $\left(\hat{\rho}_m\right)_j$ using (2.106).
    (c) end
IV end
```

The Viterbi algorithm acts similarly to the expectation step in the EM algorithm. Inference was made on the latent variable $\boldsymbol{z}_1^T$ and a most likely path given $\hat{\boldsymbol{\theta}}_{j-1}$ was computed. For this, a few intermediate calculations were required at each time step. The first, was the probability of the most likely state sequence from $t = t_1$ to t_i given the last state was state m (denoted as $\psi_i(m)$) or $z_{im} = 1$. The second, expressed as $\Psi_i(m)$, was the most likely previous state given the most likely sequence from $t = t_1$ to t_i ended in state m or $z_{im} = 1$. $\psi_i(m)$ and $\Psi_i(m)$ were calculated for all states m in the forward iteration of the Viterbi listed as (2.103). For (2.103) and (2.104), recall that $p\left(z_{1m} = 1 \middle| \hat{\theta}_{j-1}\right) = \left(\hat{\rho}_m\right)_j$ and $y_i|z_{in} = 1, \hat{\theta}_{j-1} \sim \mathcal{N}\left(\mu_n, \sigma^2\right)$.

Calculating the forward iteration of the Viterbi Path $\left(v\left(\boldsymbol{z}_1^T\right)\right)$ (2.103)

```
I. Initialization (The first timestep or $t_2$)
    i. For $m = 1 : k$
        A. $\psi_1(m) = p\left(z_{1m} = 1 \middle| \hat{\theta}_{j-1}\right) p\left(y_1 \mid z_{1m} = 1, \hat{\theta}_{j-1}\right)$
    ii. end
II. For $i = 2 : T$
    i. For $n = 1 : k$
        A. $\psi_i(n) = \left[\max_m \psi_{i-1}(m) \left(\hat{q}_{mn}\right)_{j-1}\right] p\left(y_i|z_{in} = 1, \hat{\theta}_{j-1}\right)$
        B. $\Psi_i(n) = \operatorname*{argmax}_m \psi_{i-1}(m) \left(\hat{q}_{mn}\right)_{j-1}$
    ii. end
III. end
```

Then, $\psi_T(m)$ was the probability from t_1 to t_T of the most likely path ending in state m. Therefore, by choosing m to be the state that maximized $\psi_T(m)$, $\psi_T(m)$ was the probability of the most likely path from t_1 to t_T. From this, the Viterbi Path was traced backward using $\Psi_i(m)$ from $i = T$ to $i = 1$ and was represented as $v(\boldsymbol{z}_i^T)$. The backward iteration was listed as (2.104) where m was the state of the current time (i) and n was the state at time $i + 1$.

Calculating the backward iteration of the Viterbi Path $\left(v\left(\boldsymbol{z}_1^T\right)\right)$ (2.104)

```
I. $m = \underset{r}{\operatorname{argmax}}\, \psi_T(r)$
II. $v\left(z_{Tm}\right) = 1$ and $v\left(z_{Tr}\right) = 0\ \forall\ r \neq m \in \{1, 2, .., k\}$
III. $n = m$
IV. For $i = T - 1 : 1$
    i. $m = \Psi_{i+1}(n)$
    ii. $v\left(z_{im}\right) = 1$ and $v\left(z_{ir}\right) = 0\ \forall\ r \neq m \in \{1, 2, .., k\}$
    iii. $v\left(\boldsymbol{z}_i^T\right) = \begin{bmatrix} v\left(\boldsymbol{z}_i\right) \\ v\left(\boldsymbol{z}_{i+1}^T\right) \end{bmatrix}$
    iv. $n = m$
V. end
```

As with the Baum-Welch algorithm, underflow is likely when computing $\psi_i(n)$. Therefore, the logarithm of all probabilities in (2.103) and (2.104) should be used. After the Viterbi path was computed, $\hat{\boldsymbol{\theta}}_j$ was computing (2.105), similar to the maximization step of the EM-algorithm.

$$\hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{y}_1^T, v_j(\boldsymbol{z}_1^T)|\boldsymbol{\theta}) \tag{2.105}$$

The MLE for $\boldsymbol{\theta}$ was computed by maximizing (2.88) for each parameter. They were listed as (2.106) to (2.109) where $v_j\left(\boldsymbol{z}_1^T\right)$ was the Viterbi path computed in the j^{th} iteration of the algorithm.

$$\hat{\boldsymbol{\rho}}_j = v_j\left(\boldsymbol{z}_1\right) \tag{2.106}$$

$$\left(\hat{q}_{mn}\right)_j = \frac{\sum\limits_{i=2}^{T} v_j\left(z_{(i-1)m}\right) v_j\left(z_{in}\right)_j}{\sum\limits_{i=1}^{T} v_j\left(z_{im}\right)} \tag{2.107}$$

$$\left(\hat{\mu}_m\right)_j = \frac{\sum\limits_{i=1}^{T} v_j\left(z_{im}\right) y_i}{\sum\limits_{i=1}^{T} v_j\left(z_{im}\right)} \tag{2.108}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{T}\left[\sum_{m=1}^{k} v_j\left(z_{im}\right)\left(y_i - \left(\mu_m\right)_j\right)^2\right]}{T} \tag{2.109}$$

Notice, that (2.106) to (2.109) are the same as the maximum likelihood estimations listed in Section 2.4.2 where $\gamma_i(m)$ was substituted with $v_j\left(z_{im}\right)$ and $\xi_i(mn)$ was substituted with $v_j\left(z_{im}\right) v_j\left(z_{(i+1)n}\right)$. Therefore, the derivations of (2.106) to (2.109) can be reproduced by making the aforementioned substitutions in Appendix C.3.3. Further information on the Viterbi training algorithm can be found in [24] and [44]. A good review of both the Baum-Welch algorithm and the Viterbi algorithm can be found in [43].

.

A Bayesian model for k discrete states, with k distinct Gaussian emissions

The Bayesian model employed a Gibbs sampler where $\boldsymbol{\theta}$ and $\boldsymbol{z}_1^T$ were treated as random variables. Like the Baum-Welch algorithm and Viterbi training, the Gibbs sampler can be thought of as having two parts. The first was a sampler for

$\boldsymbol{z}_1^T$, which was done using a version of the forward filter backward sampler found in [9]. The second part was sampling from $\boldsymbol{\theta}$ as found in [48]. Following the same notation of the Bayesian model for the DLM, the j^{th} draw will be denoted as $\boldsymbol{\theta}_j$ and $\left(\boldsymbol{z}_1^T\right)_j$. In addition, $\boldsymbol{\theta}_{-k}$ represents all parameters of $\boldsymbol{\theta}$ except k. An overview of the sampler for N samples was listed as (2.110).

Gibbs sampler for k states with k distinct Gaussian emissions (2.110)

```
I Pick $\boldsymbol{\theta}_0$
II for $j = 1$ to $N$
    (a) Draw $\left(\boldsymbol{z}_1^T\right)_j$ from $p\left(\boldsymbol{z}_1^T|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}\right)$ listed as (2.114)
    (b) Draw $\boldsymbol{\rho}_j$ from $p\left(\boldsymbol{\rho}|\boldsymbol{y}_1^T, \left(\boldsymbol{z}_1^T\right)_j, (\boldsymbol{\theta}_{-\rho})_{j-1}\right)$ listed as (2.115).
    (c) for $m = 1$ to $k$
        i. for $n = 1$ to $k$
            a. Draw $(q_{mn})_j$ from $p\left(q_{mn}|\boldsymbol{y}_1^T, \left(\boldsymbol{z}_1^T\right)_j, (\boldsymbol{\theta}_{-q_{mn}})_{j-1}\right)$ or (2.116).
        ii. end
        iii. Draw $(\mu_m)_j$ from $p\left(\mu_m|\boldsymbol{y}_1^T, \left(\boldsymbol{z}_1^T\right)_j, (\boldsymbol{\theta}_{-\mu_m})_{j-1}\right)$ listed as (2.117).
    (d) end
    (e) Draw $\sigma_j^2$ from $p\left(\sigma^2|\boldsymbol{y}_1^T, \left(\boldsymbol{z}_1^T\right)_j, (\boldsymbol{\theta}_{-\sigma^2})_{j-1}\right)$ listed as (2.118).
III end
```

The updates of the forward filter backward sampler had similar definitions to the updates for the DLM. The state update and state forcast were defined as (2.111) and (2.112) respectively.

$$a_i(m) = p(z_{im} = 1|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta}) \tag{2.111}$$

$$\alpha_i(m) = p(z_{im} = 1|\boldsymbol{y}_1^{i}, \boldsymbol{\theta}) \tag{2.112}$$

Since there was no observation before $\boldsymbol{y}_1$, the initial state update and state forecast were slightly different. Therefore, the state update employed $p(\boldsymbol{z_1}|\theta)$ provided in (2.88). Using $p(\boldsymbol{z}_1|\theta)$ from (2.88), the probability of state m occurring at time $t = t_1$ was ρ_m or $p(z_{1m} = 1|\boldsymbol{\theta}) = \rho_m$. From this, the forward filter algorithm was listed as 2.113, and a derivation can be found in Appendix C.4.1.1. Since the probabilities were normalized at every step, underflow was not a problem. Therefore, the algorithm can be coded as seen in (2.113).

Forward filter for k states with k distinct Gaussian emissions (2.113)

```
I for $m = 1$ to $k$
    (a) $a_1(n) = p(z_{1n} = 1|\boldsymbol{\theta})$ where $p(z_{1n} = 1|\boldsymbol{\theta}) = \rho_n$
    (b) $\alpha_1(n) = \frac{a_1(n)p(y_1|z_{1n}=1,\boldsymbol{\theta})}{\sum_{m=1}^{k} a_1(m)p(y_1|z_{1n}=1,\boldsymbol{\theta})}$
        where $y_1|z_{1m} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_m, \sigma^2)$.
II end
III for $i = 1$ to $T$
    (a) for $n = 1$ to $k$
        i. $a_i(n) = \sum_{m=1}^{k} \alpha_{i-1}(m)q_{mn}$
        ii. $\alpha_i(n) = \frac{a_i(n)p(y_i|z_{in}=1,\boldsymbol{\theta})}{\sum_{m=1}^{k} a_i(m)p(y_i|z_{im}=1,\boldsymbol{\theta})}$
            where $y_i|z_{im} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_m, \sigma^2)$.
    (b) end
IV end
```

The backward step drew $\boldsymbol{z}_i$ from $i = T$ to 1. Since draws were made in the reverse direction, $\boldsymbol{z}_{i+1}$ had already been drawn when $\boldsymbol{z}_i$ was drawn. Therefore, the definition of $\beta_i(m)$ was chosen to be $p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{\theta})$ to include the knowledge of the previous draw. The case for $\boldsymbol{z}_T$ was different as there was no $\boldsymbol{z}_{T+1}$. For the case, $i = T$, $p(z_{Tm} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}) = \alpha_T(m)$ so no additional calculation was required. This resulted in the backward sampling algorithm listed as (2.114) and was derived in Appendix C.4.1.2. In (2.114) $Cat(\alpha_1, \alpha_2, ..., \alpha_n)$ denotes a categorical distribution where the i^{th} element occurs with a probability of α_i, which is a multinomial distribution with only 1 trial.

```
The backward sample for $k$ states with $k$ distinct Gaussian emissions  (2.114)

  I Draw $\boldsymbol{z}_T$ from $Cat\left(\alpha_T(1), \alpha_T(2), ..., \alpha_T(k)\right)$

  II For $i = T$ to 1

      (a) For $m = 1$ to $k$

            i. $b_i(m) = \sum_{n=1}^{k} \alpha_i(m) q_{mn} z_{(i=1)n}$

      (b) end

      (c) For $m = 1$ to $k$

            i. $\beta_i(m) = \frac{b_i(m)}{\sum_{n=1}^{k} b_i(n)}$

      (d) end

      (e) Draw $\boldsymbol{z}_i$ from $Cat\left(\beta_i(1), \beta_i(2), ..., \beta_i(k)\right)$

  III end
```

Since this is a Bayesian Model, a prior had to be placed on $\boldsymbol{\theta}$. Given that the forward filtering backward sampling algorithm within a Gibbs sampler was computationally expensive, conditionally conjugate priors were placed on each conditional posterior. The first parameter considered was $\boldsymbol{\rho}$. Since $\boldsymbol{z}_1|\boldsymbol{\theta} \sim Cat(\boldsymbol{\rho})$, the conditional conjugate prior was $\boldsymbol{\rho} \sim Dir(\kappa_{\rho_1}, \kappa_{\rho_2}, ..., \kappa_{\rho_k})$. This resulted in the conditional posterior listed as (2.115) and was derived in Appendix C.4.2.1. The second was $\boldsymbol{q}_{m\text{-}}$, where $\boldsymbol{q}_{m-} = (q_{m1}, q_{m2}, ..., q_{mk})$. $\boldsymbol{q}_{m\text{-}}$ is equivalent to $p(\boldsymbol{z}_{i+1}|z_{im} = 1, \boldsymbol{\theta})$ in (2.88) which was a categorical as well. Therefore, the conditional conjugate prior for $\boldsymbol{q}_{m\text{-}}$ was $Dir(\kappa_{q_{m1}}, \kappa_{q_{m2}}, ..., \kappa_{q_{mk}})$. The conditional posterior was listed as (2.116), and a derivation can be found in C.4.2.2. The next parameter addressed was μ_n. Since μ_n was the mean in a normal distribution, the conditionally conjugate prior was $\mu_n \sim \mathcal{N}(m_n, \sigma_n^2)$. The conditional posterior was listed as (2.117) and a derivation can be found in Appendix C.4.2.3. Lastly, the conditionally conjugate prior for σ^2 was an inverse gamma distribution or $\sigma^2 \sim \mathcal{IG}\left(\frac{n_{\sigma^2}}{2}, \frac{d_{\sigma^2}}{2}\right)$. The conditional posterior for σ^2 was numbered (2.118) and was derived in C.4.2.4.

Priors can be chosen to limit the impact of the prior on the poster if necessary. The non-informative case for the Dirichlet distribution was to employ the symmetric Dirichlet distribution or $\kappa_1 = \kappa_2 = ... = \kappa_n$. The normal prior was previously discussed in the context of the DLM and by setting $m_n = 0$ and choosing a large σ_n^2, the information contributed by the prior was limited. Finally, to make a non-informative prior for σ^2, $n_{\sigma^2} = d_{\sigma^2} = 0$ could be used. In this case, the prior was no longer an inverse gamma distribution, but it contributes nothing to the posterior.

$$\rho|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\rho} \sim Dir(z_{11} + \kappa_{\rho_1}, z_{12} + \kappa_{\rho_2}, \ldots, z_{1k} + \kappa_{\rho_k}) \tag{2.115}$$

$$\boldsymbol{q}_{m\text{-}}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}} \sim Dir\left(\sum_{i=1}^{T} z_{i1} z_{(i-1)m} + \kappa_{q_{m1}}, ..., \sum_{i=1}^{T} z_{ik} z_{(i-1)m} + \kappa_{q_{mk}}\right) \tag{2.116}$$

$$\mu_n|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-m_n} \sim \mathcal{N}\left(\frac{\sigma_n^2 \sum\limits_{i=0}^{T} y_i z_{in} + \sigma^2 m_n}{\sigma_n^2 \sum\limits_{i=0}^{T} z_{in} + \sigma^2}, \frac{\sigma_n^2 \sigma^2}{\sigma_n^2 \sum\limits_{i=0}^{T} z_{in} + \sigma^2}\right) \tag{2.117}$$

$$\sigma^2|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\sigma^2} \sim \mathcal{IG}\left(\frac{n_{\sigma^2} + T}{2}, \frac{d_{\sigma^2} + S^2}{2}\right) \tag{2.118}$$

$$S^2 = \sum_{i=0}^{T} \left[\sum_{n=1}^{k} z_{in}(y_i - \mu_n)^2 \right]$$

As stated previously, the forward filter backward sampler can be found in [9], and the sampling of parameters can be found in [48]. However, [52] contains an overview of the entire process, derivations, and extensions. The three algorithms for making inference on $\boldsymbol{z}_1^T$ and $\boldsymbol{\theta}$ could all be applied to the models in Section 1.1. Therefore, the algorithms were not compared here but were compared in the section that discusses their performance on the experiments for the models in Section 1.1.

Chapter 3

Modeling of Biochemical States of DNA Replication Restricted to Two States

3.1 The 2-State Experiment

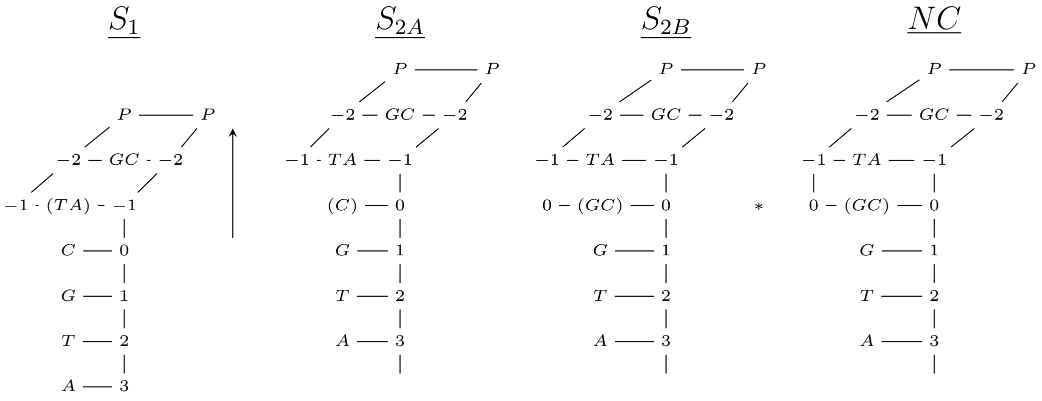


Figure 3.1: A diagram depicting a nucleotide addition cycle where the location of the active site of the DNAP is denoted by (∗)

In Chapter 1.1, methods to study the biochemical states of DNA replication were discussed. Two specific experiments were studied. The focus of this chapter was the experiment that isolated states S_1 and S_{2A}. Making inference on this experiment was equivalent to the two state, two signals continuous time Markov chain indirectly measured at discrete times discussed in Section 2.4.2.

An $\alpha-$hemolysin nanopore, which is a very small channel, was employed to study the biochemical states of DNA replication. Dahl and his colleagues found that changes in biochemical state could be tracked when a single strand of DNA attached to a DNA polymerase (a catalyst to replication) was hung through the nanopore while an electrical current was conducted through the nanopore [11]. In an experiment designed by Lieberman and her colleagues, state S_1, the pre-translocation state, and state S_{2A}, the post-translocation state were isolated. Therefore, the DNA and DNA polymerase binary complex vacillated between state S_1 and state S_{2A} in Figure 3.1. In addition, the dwell times in each state were exponential and could be associated with different currents [31]. This resulted in a continuous time Markov chain as discussed in Section 2.4.1.

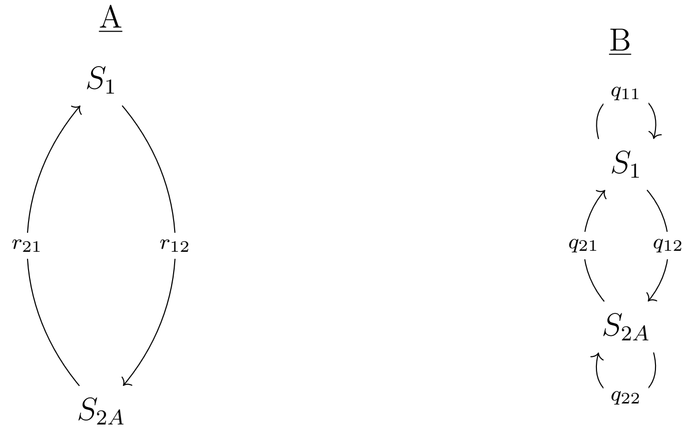


Figure 3.2: State diagrams of the continuous time 2-state system with transition rates r_{ij} (A) and discrete time 2-state system with transition probabilities q_{ij} (B)

The state diagram for the continuous system is in Figure 3.2.A where r_{mn} was the transition rate from state m to n. The corresponding discrete time system is pictured in Figure 3.2.B where q_{mn} was the transition probability from state m to n. Inference on r_{mn} was the subject of this study, but it could not be made directly. Therefore, inference was made on q_{mn} in the discrete time system and this was transformed to r_{mn} using the equalities in (3.1)-(3.5) where $q_{11} = 1 - q_{12}$ and $q_{22} = 1 - q_{21}$. For further discussion on (3.1)-(3.5), see Section 2.4.1.

$$\begin{aligned} q_{12} =& \frac{r_{12}}{r_{12} + r_{21}} \left(1 - \exp(-\Delta t(r_{12} + r_{21}))\right) && (3.1) \\ q_{21} =& \frac{r_{21}}{r_{12} + r_{21}} \left(1 - \exp(-\Delta t(r_{12} + r_{21}))\right) && (3.2) \\ r_{12} + r_{21} =& - \frac{\ln(1 - [q_{12} + q_{21}])}{\Delta t} && (3.3) \\ r_{12} =& \frac{q_{12}(r_{12} + r_{21})}{1 - \exp(-\Delta t(r_{12} + r_{21}))} && (3.4) \\ r_{21} =& \frac{q_{21}(r_{12} + r_{21})}{1 - \exp(-\Delta t(r_{12} + r_{21}))} && (3.5) \end{aligned}$$

The observed data was the amplitude of the electric current passing through the nanopore. The current at time t_i denoted as y_i was measured at $t = (t_1, ..., t_T)$ where $\Delta t = t_i - t_{i-1}$ for all $i \in \{2, 3, ..., T\}$. The unobserved state at time t_i was represented by $\boldsymbol{z}_i$ where $\boldsymbol{z_i} = (1, 0)'$ if the system was in state S_1 and $\boldsymbol{z_i} = (0, 1)'$ if the system was in state S_{2A}. Since it was observed by Liebermen and her colleagues that the different electric signals could be associated with different states, $y_i \sim \mathcal{N}(\mu_{S_1}, \sigma^2)$ if $\boldsymbol{z_i} = (1, 0)'$ and $y_i \sim \mathcal{N}(\mu_{S_{2A}}, \sigma^2)$ if $\boldsymbol{z_i} = (0, 1)'$. The joint likelihood of $\boldsymbol{y_1^T} = (y_1, ..., y_T)$ and $\boldsymbol{z_1^T} = (z_1, ..., z_T)$ was listed as (3.6) and an explanation can be found in Section 2.4.1.

$$p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta}) \text{ for 2 states} \quad (3.6)$$

$$\begin{aligned}
p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta}) &= p(\boldsymbol{z}_1, \boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) &= p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)1} = 1)^{\boldsymbol{z}_{(i-1)1}} p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)2} = 1)^{\boldsymbol{z}_{(i-1)2}} \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)1} = 1) &= (q_{11})^{z_{i1}} (q_{12})^{z_{i2}} \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)2} = 1) &= (q_{21})^{z_{i1}} (q_{22})^{z_{i2}} \\
p(\boldsymbol{z}_1|\boldsymbol{\theta}) &= \rho_1^{z_{11}} \rho_1^{z_{12}} \\
y_i|\boldsymbol{z}_{im} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_m, \sigma^2) \\
p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) &= [p(y_i|\boldsymbol{z}_{i1} = 1, \boldsymbol{\theta})]^{z_{i1}} \, [p(y_i|\boldsymbol{z}_{i2} = 2, \boldsymbol{\theta})]^{z_{i2}}
\end{aligned}$$

3.2 Parameter Estimation

3.2.1 Methology

The indirect discrete measurement of state was used to study the transition rates from S_1 to S_{2A} and S_{2A} to S_1. Therefore, inference on the transition probabilities was made using three different methods and subsequently mapped to the continuous transition rates.

One method, called the Baum-Welch algorithm, maximizes point estimates for the parameters of the discrete system iteratively. The Baum-Welch Algorithm is recognized as a special case of the Expectation Maximization (EM) Algorithm, but it was developed by Leonard Baum and his colleagues separately and before the EM algorithm was introduced by Dempster [4] [1]. The Baum-Welch algorithm has been studied extensively in a number of applications. A similar application of the Baum-Welch was discussed in "Characterization of Single Channel Currents Using Digital Signal Processing Techniques Based on Hidden Markov Models". In this case, Chung and his colleagues used the Baum-Welch algorithm to measure single channel ion currents. They were able to infer transition rates with a high degree of accuracy when the noise (σ) was up to 0.8 the difference between signal levels [10].

Another method that maximizes point estimates for the parameters of the discrete system is known as Viterbi extraction, Viterbi training, or segmental k-means [24] [44]. For comparison to the Baum-Welch algorithm, Qin studied applying the second algorithm which he refers to as the segmental k-means algorithm [42] to a situation similar to [10]. A good comprehensive review of both methods was written by Rabiner [43]. The last method applied a Gibbs sampler for the posterior of the parameters. This model has also been studied extensively in a Bayesian Paradigm where this model can also be referred to a Markov switching model. General models of this type have been discussed in articles, textbooks, and reviews [9] [52] [53] [48]. The details of each algorithm were discussed in detail in Section 2.4.2.

3.2.2 Datasets

The Baum-Welch algorithm, Viterbi Training algorithm, and the Bayesian Gibbs sampler were evaluated by computer generated idealized datasets. This provided a simple way to assess the performance of the algorithm and an additional way to confirm convergence. Since the transition rates are the focus of this study, the dataset was created by simulating a continuous time Markov process with known transition rates. This was done using the Gillespie algorithm [19] and then adding additional noise to simulate the electric current. For these datasets, the complex was arbitrarily started in state S_1, but this could also be randomized. The current state of the CTMC was denoted as $X(t) = 1$ if the system was in state S_1 at time t and $X(t) = 2$ if the system was in state S_{2A} at time t. Then $X(t)$ was generated using the algorithm below, mapped to $\boldsymbol{z}_1^T$ which in turn was used to generate $\boldsymbol{y}_1^T$. To generate $X(t)$, let E_j be the time of the j^{th} event or transition and ΔE_j be $E_j - E_{j-1}$. Since the complex started in state S_1, if j is odd, a change from S_1 to S_{2A} occurred at time E_j. If j is even, a change from S_{2A} to S_1 occurred at time E_j.

The Gillespie algorithm for simulating X(t) for the 2-state system

```
I. Set j=1 and t=0.
II. Generate $E_1 \sim \exp(r_{12})$
III. $t = E_1$, $j = j + 1$
IV. while $t < t_T$
    A. if $j$ is odd
        i. Generate $\Delta E_j \sim \exp(r_{12})$
        ii. $E_j = \Delta E_j + E_{j-1}$ and $t = E_j$
    B. if $j$ is even
        i. Generate $\Delta E_j \sim \exp(r_{21})$
        ii. $E_j = \Delta E_j + E_{j-1}$ and $t = E_j$
    C. $j = j + 1$
```

From this, $X(t) = 1$ for$[0, E_1]$, $[E_2, E_3]$, $[E_4, E_5]$,.... and $X(t) = 2$ for $[E_1, E_2]$, $[E_3, E_4]$, $[E_5, E_6]$,.... Since $\boldsymbol{z}_1^T$ was measured at $\boldsymbol{t}_1^T = (t_1, t_2, ..., t_T)$, $\boldsymbol{z}_i$ was set equal to the value associated with $X(t_i)$. Finally, y_i was drawn from $y_i \sim p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})$.

3.2.3 Convergence

To asses the convergence of the inference on the transition rates of the three algorithms, they were run on a dataset generated from the Gillespie algorithm as listed in 3.2.2. The first dataset was generated with $r_{12} = 600$, $r_{21} = 2000$, $\mu_{S_1} = 32$, $\mu_{S_{2A}} = 26$, and $\sigma = 1.5$ where $\Delta t = \frac{1}{10000}$.

Trial Name	Run 1	Run 2	Run 3
$(\hat{q}_{12})_0$	0.092	0.034	0.058
$(\hat{q}_{21})_0$	0.027	0.015	0.078
$\hat{\sigma}_0^2$	4.354	1.127	0.679
$(\hat{\mu}_{S_1})_0$	32.09	30.62	33.26
$(\hat{\mu}_{S_{2A}})_0$	20.31	20.31	20.31

Table 3.1: Initial values for runs of the Baum-Welch algorithm

Although convergence was checked for all simulations, it will be discussed in more detail for a 0.5 second dataset with the aforementioned parameters. To asses convergence, each algorithm was tested using an overdispersed initial guess, compared against MLE estimates when $\boldsymbol{z}_1^T$ was known, and compared with each other. First, the Baum-Welch algorithm was tested using randomly generated numbers as initial guesses which were denoted as $\hat{\boldsymbol{\theta}}_0$. Using randomly generated values for $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$, it was discovered that the Baum-Welch algorithm does not converge to the global minimum for all cases. If μ_{S_1} and $\mu_{S_{2A}}$are chosen such that $\mu_{S_1} > \bar{\boldsymbol{y}}_0^T$ and $\mu_{S_{2A}} > \bar{\boldsymbol{y}}_0^T$ where $\bar{\boldsymbol{y}}_0^T$ is the mean of the observed currents then the algorithm converges to a local minimum $\mu_{S_1} \approx \mu_{S_{2A}} \approx \bar{\boldsymbol{y}}_0^T$. The same also occurred if $\mu_{S_1} < \bar{\boldsymbol{y}}_0^T$. and $\mu_{S_{2A}} < \bar{\boldsymbol{y}}_0^T$. Therefore, the same experiment was run where the initial guesses were $\mu_{S_1} = \max\left(\boldsymbol{y}_0^T\right)$ and $\mu_{S_{2A}} = \min\left(\boldsymbol{y}_0^T\right)$ and the rest of the parameters were randomly generated. In Figure (3.3), $\hat{k}_j$ represents the Baum-Welch algorithm estimate of k after j iterations of the algorithm. The values of $(\hat{q}_{12})_j$, $(\hat{q}_{21})_j$, $\hat{\sigma}_j^2$, $(\hat{\mu}_{S_1})_j$, and $(\hat{\mu}_{S_{2A}})_j$ for the runs in Table 3.1 were compared in Figure 3.3.

Although convergence was checked for all simulations, it will be discussed in more detail for a 0.5 second dataset with the aforementioned parameters. To asses convergence, each algorithm was tested using an over dispersed initial guess, compared against MLE estimates when $\boldsymbol{z}_1^T$ was known, and compared with each other. First, the Baum-Welch algorithm was tested using randomly generated numbers as intial guesses which were denoted as $\hat{\boldsymbol{\theta}}_0$. Using randomly generated

values for $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$, it was discovered that the Baum-Welch algorithm does not converge to the global minimum for all cases. If μ_{S_1} and $\mu_{S_{2A}}$are chosen such that $\mu_{S_1} > \bar{\boldsymbol{y}}_0^T$ and $\mu_{S_{2A}} > \bar{\boldsymbol{y}}_0^T$ where $\bar{\boldsymbol{y}}_0^T$ is the mean of the observed currents then the algorithm converges to a local minimum $\mu_{S_1} \approx \mu_{S_{2A}} \approx \bar{\boldsymbol{y}}_0^T$. The same also occurred if $\mu_{S_1} < \bar{\boldsymbol{y}}_0^T$. and $\mu_{S_{2A}} < \bar{\boldsymbol{y}}_0^T$. Therefore, the same experiment was run where the initial guesses were $\mu_{S_1} = \max\left(\boldsymbol{y}_0^T\right)$ and $\mu_{S_{2A}} = \min\left(\boldsymbol{y}_0^T\right)$ and the rest of the parameters were randomly generated. In Figure (3.3), $\hat{k}_j$ represents the Baum-Welch algorithm estimate of k after j iterations of the algorithm. The values of $\left(\hat{q}_{12}\right)_j$, $\left(\hat{q}_{21}\right)_j$, $\hat{\sigma}_j^2$, $\left(\hat{\mu}_{S_1}\right)_j$, and $\left(\hat{\mu}_{S_{2A}}\right)_j$ for the runs in Table 3.1 were compared in Figure 3.3.

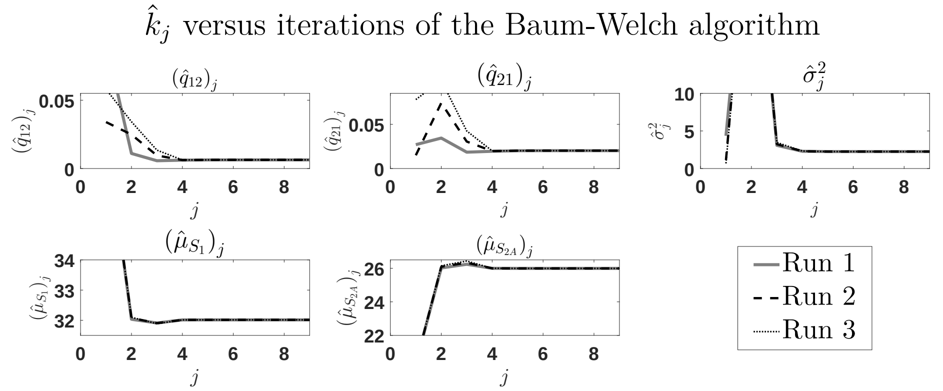


Figure 3.3: Values of $\left(\hat{q}_{12}\right)_j$, $\left(\hat{q}_{21}\right)_j$, $\hat{\sigma}_j^2$, $\left(\hat{\mu}_{S_1}\right)_j$, and $\left(\hat{\mu}_{S_{2A}}\right)_j$ vs number of iterations (j)

In each simulation $\hat{q}_{12}$, $\hat{q}_{21}$, $\hat{\mu}_{S_1}$, $\hat{\mu}_{S_{2A}}$, and $\hat{\sigma}^2$ converged to 0.0063, 0.0202, 32.0144, 25.9925, and 2.2567. For reference, the true values of q_{12} and q_{21} when $r_{12} = 600$ and $r_{21} = 2000$ are approximately 0.0060 and 0.0198 while $\mu_{S_1} = 32$, $\mu_{S_{2A}} = 26$, and $\sigma^2 = 2.25$.

The same test was run on using the Viterbi training or segmental k-mean algorithm. In Figure 3.4, the results of the Viterbi training algorithm were compared for the over dispersed $\hat{\boldsymbol{\theta}}_0$ from Table 3.1. It is important to note that not all simulations took 7 iterations to converge. To better compare $\hat{\boldsymbol{\theta}}$ for all three runs, the traces of $\hat{\boldsymbol{\theta}}$ for the trials that converge more quickly were extended to 7. Unlike the Baum-Welch, the Viterbi Training algorithm did not converge to the same value for all $\hat{\boldsymbol{\theta}}_0$ from Table 3.1. This was particularly difficult because unlike the Baum-Welch algorithm, it was not immediately obvious that $\hat{\boldsymbol{\theta}}$ was an unwanted local minimum.

$\hat{k}_j$ versus iterations of the Viterbi Training algorithm for runs 1-3

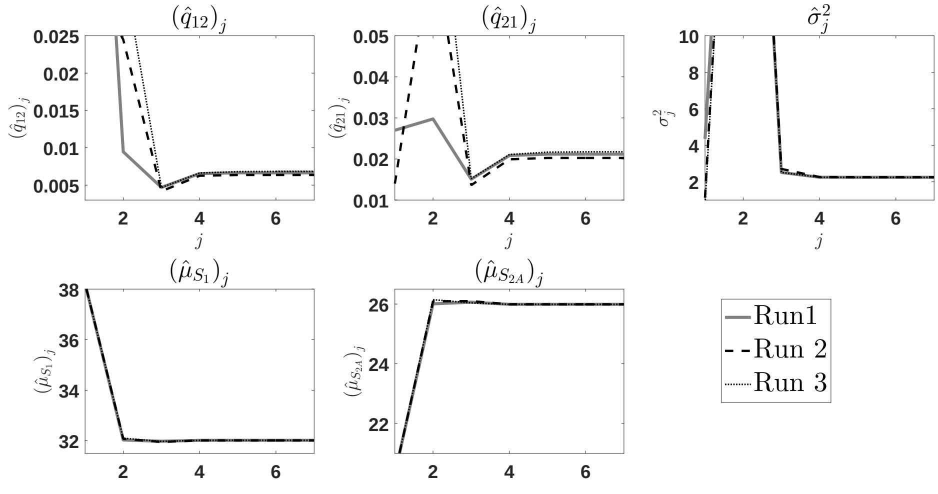


Figure 3.4: Values of $(\hat{q}_{12})_j$, $(\hat{q}_{21})_j$, $\hat{\sigma}^2_j$, $(\hat{\mu}_{S_1})_j$, and $(\hat{\mu}_{S_{2A}})_j$ vs number of iterations (j)

$\hat{\boldsymbol{\theta}}$ for runs 1-3 in were placed in Table 3.2. However, the initial guesses for $\hat{\boldsymbol{\theta}}_0$ were intentionally obtuse or overdispersed. Therefore, the Viterbi Training algorithm was retested with initial guesses that were slightly more reasonable. Table 3.3 has $\hat{\boldsymbol{\theta}}_0$ for Run 4-6 where $(\hat{q}_{12})_j$ and $(\hat{q}_{21})_j$ were within an order magnitude of $\boldsymbol{\theta}$. The results of the Viterbi training algorithm were placed in Figure 3.5. For $\hat{\boldsymbol{\theta}}_0$ from Table 3.3, the Viterbi training algorithm converged to $\hat{q}_{12} = 0.0063$, $\hat{q}_{21} = 0.0199$, $\hat{\sigma}^2 = 2.2550$, $\hat{\mu}_{S_1} = 32.0147$, and $\hat{\mu}_{S_{2A}} = 25.9926$. Following this, the Viterbi training algorithm also exhibits convergence for reasonable $\hat{\boldsymbol{\theta}}_0$. However,

$\hat{\boldsymbol{\theta}}_0$ must be more carefully chosen as the Viterbi training algorithm was more likely to get caught in undesirable local minimums that were not obviously wrong.

Trial Name	Run 1	Run 2	Run 3
$\hat{q}_{12}$	0.0066	0.0064	0.0068
$\hat{q}_{21}$	0.0212	0.0203	0.0217
$\hat{\sigma}^2$	2.2454	2.2522	2.2420
μ_{S_1}	32.016	32.015	32.017
$\hat{\mu}_{S_{2A}}$	25.990	25.992	25.990

Table 3.2: Values of $\hat{\boldsymbol{\theta}}$ using the Viterbi training algorithm for Runs 1-3

Trial Name	Run 4	Run 5	Run 6
$(\hat{q}_{12})_0$	0.0050	0.0040	0.0080
$(\hat{q}_{21})_0$	0.0210	0.0220	0.0150
$\hat{\sigma}_0^2$	1	2	3
$(\hat{\mu}_{S_1})_0$	36	40	40
$(\hat{\mu}_{S_{2A}})_0$	20	16	10

Table 3.3: Initial values for additional runs of the Viterbi training algorithm

$\hat{k}_j$ versus iterations of the Viterbi Training algorithm for runs 4-6

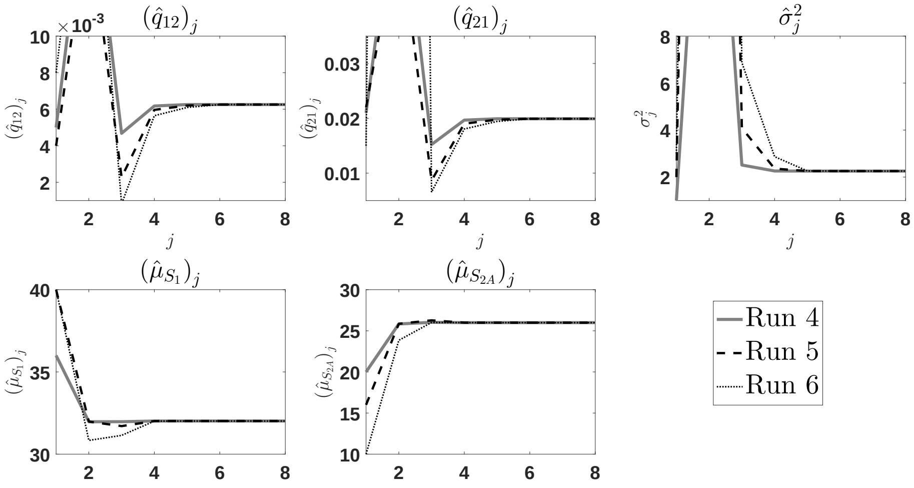


Figure 3.5: Values of $(\hat{q}_{12})_j$, $(\hat{q}_{21})_j$, $\hat{\sigma}_j^2$, $(\hat{\mu}_{S_1})_j$, and $(\hat{\mu}_{S_{2A}})_j$ vs number of iterations (j)

The Gibbs sampler was also tested by choosing an over-dispersed initial guess denoted as $\boldsymbol{\theta}_0$. As with Baum-Welch, there are complications for $\mu_{S_1} > \bar{\boldsymbol{y}}_0^T$ and

$\mu_{S_{2A}} > \bar{\boldsymbol{y}}_0^T$ or $\mu_{S_1} < \bar{\boldsymbol{y}}_0^T$ and $\mu_{S_{2A}} < \bar{\boldsymbol{y}}_0^T$. Therefore, $\mu_{S_1} > \bar{\boldsymbol{y}}_0^T$ and $\mu_{S_{2A}} < \bar{\boldsymbol{y}}_0^T$ were used in $\boldsymbol{\theta_0}$. The values for $\boldsymbol{\theta}_0$ are in Table 3.4. The first 50 draws from $q_{12}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-q_{12}}$, $q_{21}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-q_{21}}$, $\sigma^2|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\sigma^2}$, $\mu_{S_1}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\mu_{S_1}}$, and $\mu_{S_{2A}}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\mu_{S_{2A}}}$ are in Figure 3.6 where $\boldsymbol{\theta}_{-k}$ denotes all $\boldsymbol{\theta}$ except parameter k.

A Trace of the draws 1-50 from the posterior of $\boldsymbol{\theta}$

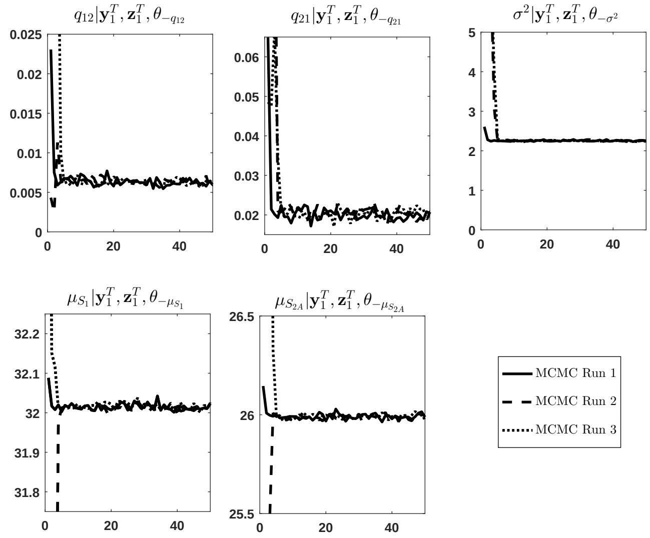


Figure 3.6: The first 50 draws from the posterior

In Figure 3.6, it can be seen that the posterior quickly approaches similar ranges for each posterior. Since the region of interest is small in 3.6, samples $4800 - 5000$ were compared from Run 1 and Run 2 for q_{12} and q_{21}. In Figure 3.7, it can be seen that both sequences appear to be stationary. Furthermore, both have converged to the same region.

Trial	MCMC	MCMC	MCMC
Name	Run 1	Run 2	Run 3
q_{12}	0.065	0.838	0.049
q_{21}	0.023	0.004	0.017
σ^2	2.608	13.77	25.15
μ_{S_1}	32.09	30.62	33.26
$\mu_{S_{2A}}$	26.15	22.66	27.84

Table 3.4: Initial values for runs of the Gibbs sampler

The last two tests were placed in Fig-

A Trace of the draws $4800 - 5000$ from the posterior of q_{12} and q_{21}

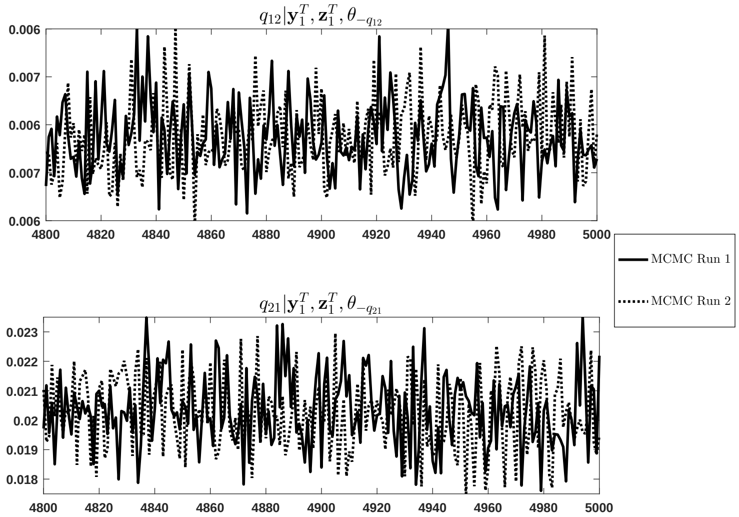


Figure 3.7: Twenty draws of q_{12} and q_{21} from the posterior

ure 3.8. The simulated posterior was compared with the point estimates from the Baum-Welch and Viterbi training algorithm. In addition, all three of the former were compared with the true value of $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ when the true values of $\boldsymbol{z}_1^T$ where known. This was done with a simulation that was 0.5 seconds long, $15,000$ draws from the posterior were simulated, and the values for $\boldsymbol{\theta}_0$ of the Gibbs sampler were generated by the Baum-Welch algorithm.

As seen in Figure 3.8, the Baum-Welch algorithm and Viterbi training algorithm both produced results close to $\hat{\boldsymbol{\theta}}$ when the true values of $\boldsymbol{z}_1^T$ where known. In addition, the results were within reasonable error of the true value of $\boldsymbol{\theta}$. The posterior of $\boldsymbol{\theta}$ was also centered near the other two algorithms as well as $\hat{\boldsymbol{\theta}}$ when the true values of $\boldsymbol{z}_1^T$ where known. Given that, plus the performance on overdispersed initial guesses, the algorithms seem to be converging to desired values. Although the Viterbi training algorithm was more sensitive to choices of initial guesses, all performed as desired with reasonable starting points.

Estimates of the three algorithms with the true $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ when $\boldsymbol{z}_1^T$ was known

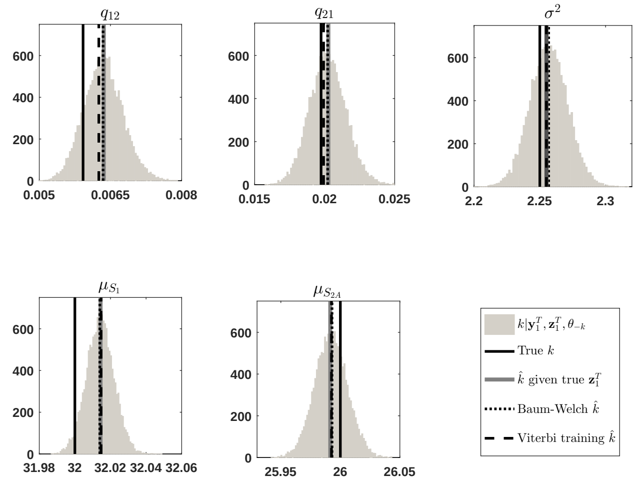


Figure 3.8: The histograms of the simulated posteriors, the points estimates, and true values of $\boldsymbol{\theta}$

3.2.4 Performance

The goal was to make inference on the transition rates of the two state, two signal system where the state was indirectly measured at discrete times. Therefore, inference using the Baum-Welch algorithm, the Viterbi training algorithm, and the Gibbs sampler was studied. To understand the performance of the different methods, different tests were used. Since point estimates provide less information about each dataset, more datasets were used to test the Baum-Welch and Viterbi training algorithms. Therefore, twenty datasets each that were one half of a second, 1 second, 2 seconds and 4 seconds long were generated. The Viterbi training algorithm and the Baum-Welch algorithm were run on each dataset (80 total datasets) to evaluate the point estimates. The nature of posterior distribution provides much more information. Therefore, it was not necessary or practical to evaluate 80 datasets as was done with the point estimates. To evaluate the Gibbs sampler for this data, one dataset was run that was four seconds long. Than the Gibbs sampler was run on the first half second, 1 second, 2 seconds and the entire dataset.

To test the Viterbi training algorithm and the Baum-Welch algorithm, each dataset was generated using (3.2.2) with $r_{12} = 600$, $r_{21} = 2000$, $\mu_{S_1} = 32$, $\mu_{S_{2A}} = 26$, $\sigma = 1.5$, and $\Delta t = \frac{1}{100000}$ seconds. For each dataset, both algorithms were run. The estimates of each algorithm minus the MLE of each dataset given the true value of $\boldsymbol{z}_1^T$ were compared. The aforementioned differences were denoted at $\hat{k} - \hat{k}$ given $\boldsymbol{z}_1^T$ and were plotted in Figure 3.9.

Estimates of the Baum Welch and Viterbi training algorithm minus the MLE of each paramters given the true $\boldsymbol{z}_1^T$

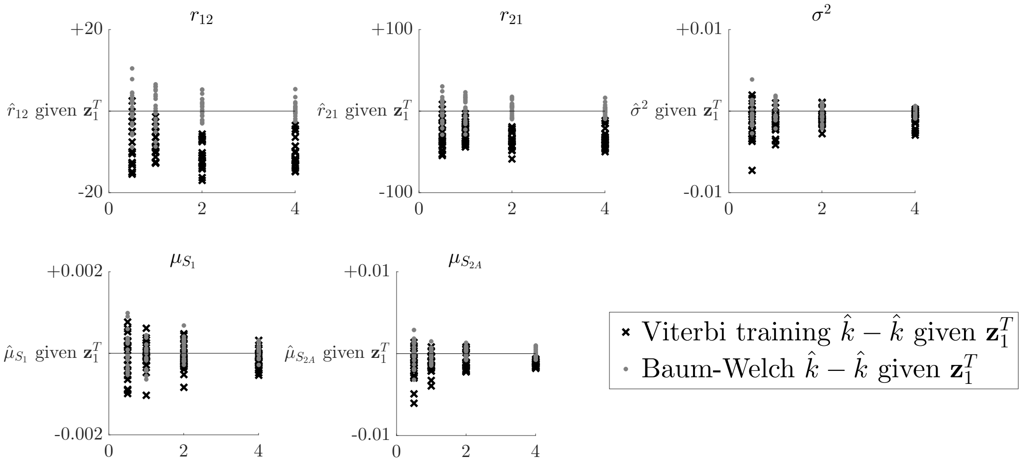


Figure 3.9: $\hat{k} - \hat{k}$ given $\boldsymbol{z}_1^T$ versus time for the Baum-Welch and Viterbi training algorithms

For σ^2, μ_{S_1}, and $\mu_{S_{2A}}$, both methods exhibited small relative error. However, Viterbi Training exhibited larger errors that appear to be biased for r_{12} and r_{21}. In fact, it has been noted by Qin that Viterbi training is not asymptotically unbiased [42] and this was observed in the empirical evidence in Figure 3.9. For the case studied, the bias was small relative to the values of r_{12} and r_{21}, but parameters that were harder to learn could amplify the aforementioned issues. To study this further, all three algorithms were run on a dataset one half of a second long where $r_{12} = 600$, $r_{21} = 2000$, $\mu_{S_1} = 32$, $\mu_{S_{2A}} = 26$, and $\sigma = 3$. The results were compared in Figure 3.10.

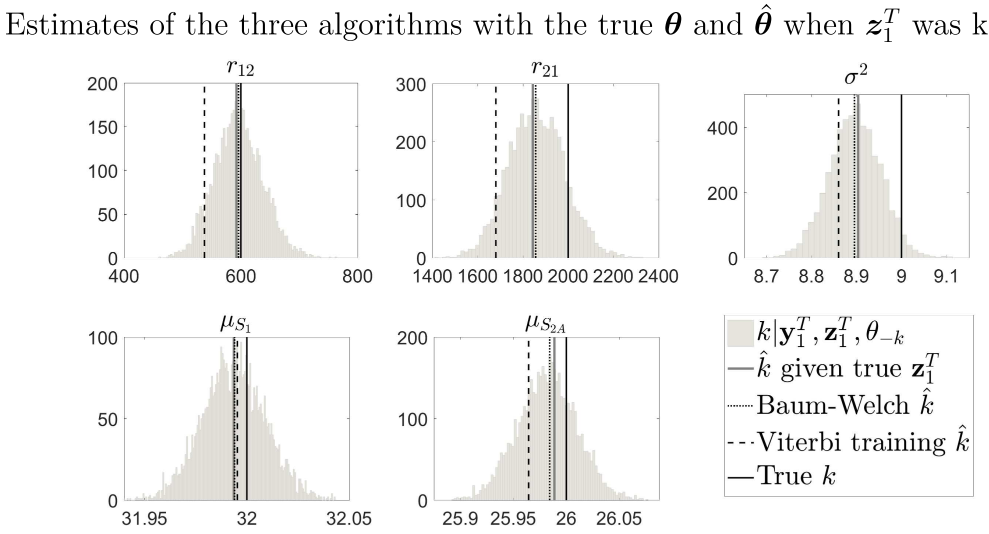


Figure 3.10: The histograms of the simulated posteriors, the points estimates, and true values of $\boldsymbol{\theta}$

As with Figure 3.8, the Baum-Welch algorithm and the MLE when $\boldsymbol{z}_1^T$ was known are very similar. However, the biased behavior of the Viterbi training was exacerbated when the value of σ was increased.

For the datasets ran, there were examples where the relative error introduced by the Viterbi training algorithm was negligible. However, this was not the case for all datasets. Furthermore, the longest iteration of the Baum-Welch algorithm for $400,000$ data points was less than 5 minutes. Although this was considerably longer than the Viterbi training algorithm, for datasets of this size, runtimes differences were inconsequential for this research. Therefore, since σ^2 was not known, the Baum-Welch algorithm would be a better tool given the dataset was not significantly larger than the data explored here.

The Gibbs sampler provides more information. Therefore, it was not necessary to run the algorithm on 80 different datasets. To explore the inference and uncertainty in the case of $r_{12} = 600$, $r_{21} = 2000$, $\mu_{S_1} = 32$, $\mu_{S_{2A}} = 26$, and $\sigma = 1.5$, one dataset was generated that was 4 second long with $\Delta t = \frac{1}{100000}$ seconds. Then MCMC simulations were run on the first half of a second, 1 second, 2 seconds, and the entire dataset. Simulations were compared in a similar fashion as in Section 2.4.1. As before, define $\left(w_{ij}\right)_k$ as the k^{th} draw from the posterior and $\left\{\left(w_{ij}\right)_k\right\}$ as the set of all draws made from the posterior. Let $\left(\epsilon_{w_{ij}}\right)_k$ be defined such that $\left(\epsilon_{w_{ij}}\right)_k + \left(w_{ij}\right)_k = w_{ij}$ where w_{ij} denotes the true value of w_{ij}. Finally, similar to $\left\{\left(w_{ij}\right)_k\right\}$, $\left\{\left(\epsilon_{w_{ij}}\right)_k\right\}$ denotes the set of all draws of $\left(\epsilon_{w_{ij}}\right)_k$ from the posterior.

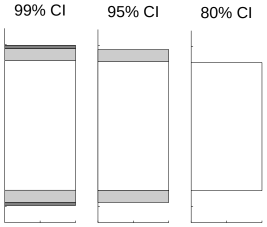


Figure 3.11: CI Demarcation

Again, the distributions were compared by plotting the credible intervals of $\left\{\left(\epsilon_{w_{ij}}\right)_k\right\}$ for each trial on the same axis. Here, the 99%, 95%, and 80% CIs (Credible Intervals) were compared and demarcated by the color schemes in Figure 3.11. The comparisons of $\{(\epsilon_{r_{12}})_k\}$, $\{(\epsilon_{r_{21}})_k\}$, $\{(\epsilon_{\sigma^2})_k\}$, $\left\{\left(\epsilon_{\mu_{S_1}}\right)_k\right\}$, and $\left\{\left(\epsilon_{\mu_{S_{2A}}}\right)_k\right\}$ were placed in Figure 3.12. The y-axis represented the size of the credible intervals of $\left\{\left(\epsilon_{w_{ij}}\right)_k\right\}$ while the x-axis represented the total time of each simulation. In Figure 3.12, the credible intervals seem to be centered on the true values for each parameter and the credible interval shrink around the true values as total time goes to ∞. As with the Baum-Welch algorithm, there was not a discernible bias for the datasets used. The credible intervals for $\{(\epsilon_{\sigma^2})_k\}$, $\left\{\left(\epsilon_{\mu_{S_1}}\right)_k\right\}$, and $\left\{\left(\epsilon_{\mu_{S_{2A}}}\right)_k\right\}$ were small relative to the true values and therefore little uncertainty was introduced even with a dataset 0.5 seconds long. The same was not true for r_{12} and r_{21}. Recall the values of r_{12} and r_{21} were 600 and 2000, so the credible intervals for the posteriors are large relative to true values for the half-second trial.

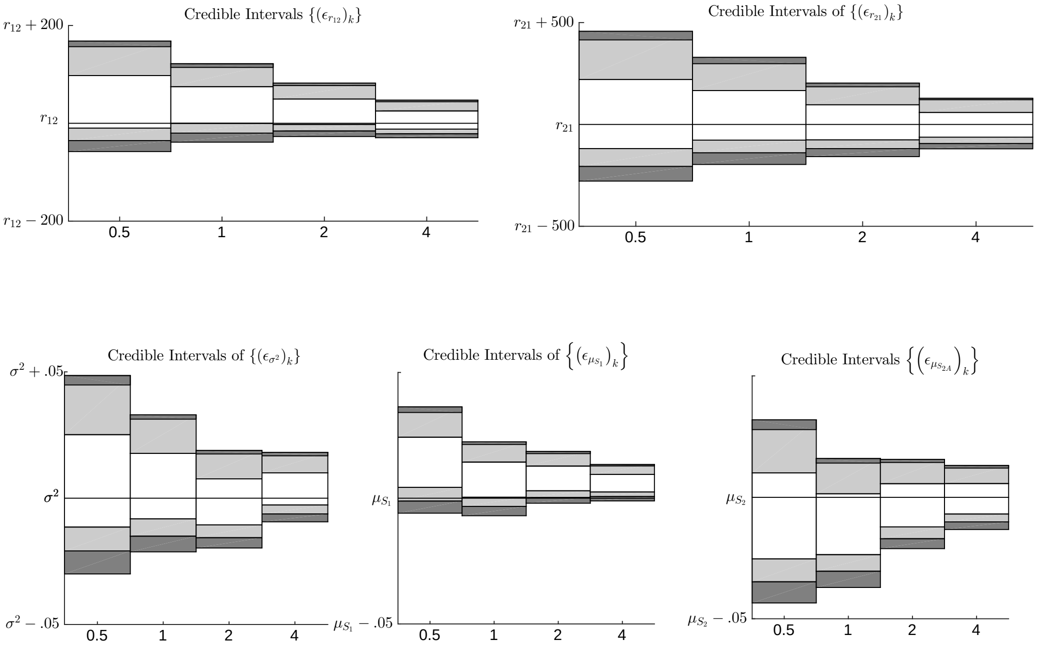


Figure 3.12: The credible intervals of $\{(\ \epsilon_{w}\ )_k\}$

To investigate the size of the credible intervals, $DQ\left(p; \left(\epsilon_{w_{ij}}\right)_k\right)$ was used as in Section 2.4.1.

$$DQ\left(p; \left(\epsilon_{w_{ij}}\right)_k\right) = \frac{1}{w_{ij}}\left[Q\left(1 - \frac{1-p}{2}; \left\{\left(\epsilon_{w_{ij}}\right)_k\right\}\right) - Q\left(\frac{1-p}{2}; \left\{\left(\epsilon_{w_{ij}}\right)_k\right\}\right)\right] \quad (3.7)$$

As a reminder, $DQ\left(p; \left(\epsilon_{w_{ij}}\right)_k\right)$ was redefined in (3.7) and $Q\left(n; \left\{\left(\epsilon_{w_{ij}}\right)_k\right\}\right)$ was defined as the quantile function of the simulated distribution of $\left\{\left(\epsilon_{w_{ij}}\right)_k\right\}$ evaluated at n. To evaluate the uncertainty against simulation time, $DQ\left(p; \left(\epsilon_{r_{12}}\right)_k\right)$

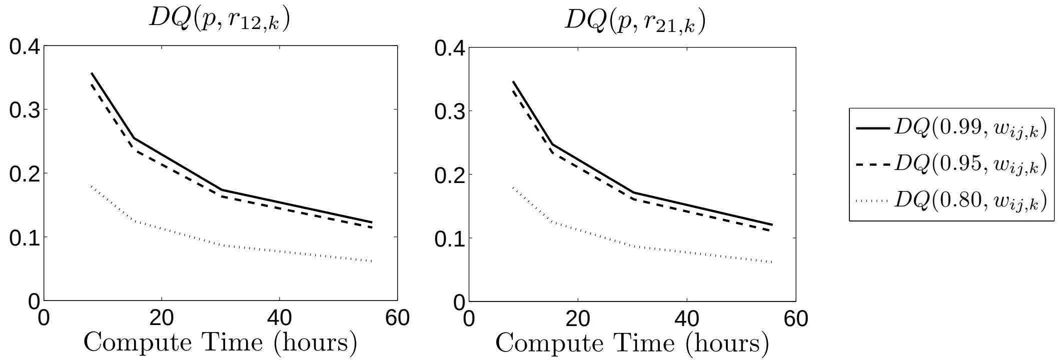


Figure 3.13: $DQ\left(p;\{w_{ij,k}\}\right)$ vs. compute time for r_{12} and r_{21}

and $DQ\left(p;\left(\epsilon_{r_{21}}\right)_k\right)$ were plotted against compute time in Figure 3.13 for $p = 99\%$, $p = 95\%$, and $p = 80\%$.

In Figure 3.13, both 99% and 95% intervals have large uncertainty for small datasets. For a dataset with a total time of four seconds, $DQ\left(0.99;\{\left(\epsilon_{r_{12}}\right)_k\}\right)$ and $DQ\left(0.99;\{\left(\epsilon_{r_{21}}\right)_k\}\}\right)$ were under 0.15. Following this, with long simulation times, $DQ\left(0.99;\{\left(\epsilon_{r_{12}}\right)_k\}\right) \approx 0.10$ and $DQ\left(0.99;\{\left(\epsilon_{r_{21}}\right)_k\}\right) \approx 0.10$ could be achieved. However, if only point estimates are desired, the Baum-Welch Algorithm was able to converge for a dataset four seconds long in under ten minutes. The choice between the two algorithms depends on the desire to describe ranges for the parameters versus the cost of extra simulation time. Finally, if the Gibbs sampler was used on a long dataset, given the nature of a Gibbs sampler, parallelization is possible.

Chapter 4

Modeling of Biochemical States of DNA Replication Restricted to Three States

4.1 The 3-State Experiment

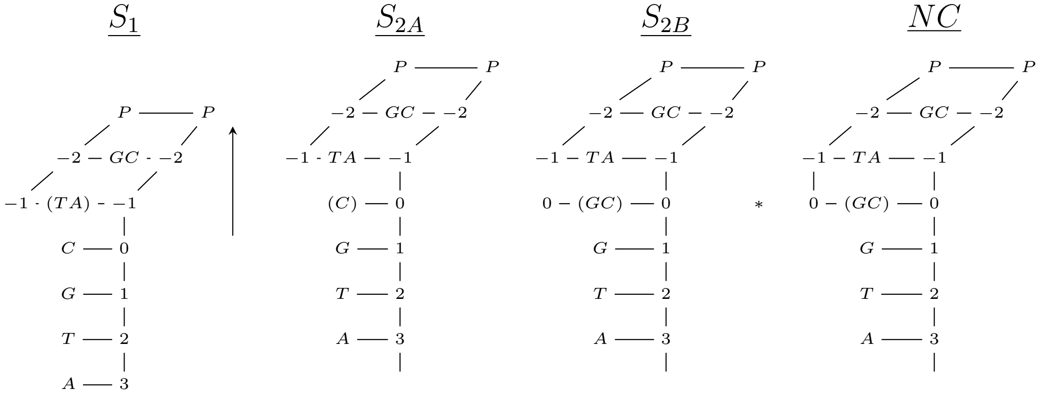


Figure 4.1: A diagram depicting a nucleotide addition cycle where the location of the active site of the DNAP is denoted by (∗)

The inference on a system with three states with three distinct signals would be almost identical to the inference explored in previous sections. However, the three state experiments studied for this research only had two distinct signals. Therefore, the inference was more difficult and previously discussed modeling techniques could not be applied.

Chapter 3 discussed an experiment designed by Lieberman and her colleagues. In that experiment, the biochemical states of the DNA were tracked by hanging the DNA and DNAP complex through a nanopore and applying an electric current. This experiment was designed such that states S_1 and S_{2A} in Figure 4.1 [31] were isolated. In a later experiment, Lieberman and her colleagues isolated states S_1, S_{2A}, and S_{2B} [32]. For this experiment, $dNTP$, the necessary complement for the addition cycle, was added to the solution in addition to the DNA and DNAP binary complex. Then the DNA and DNAP could move from the pre-translocation state labeled as S_1, to the post translocation state (S_{2A}), and the complementary base pair could associate making the ternary complex pictured as state S_{2B}. The experiment was engineered such that the $dNTP$ could not form a covalent bond with the backbone of the DNA. Following this, the $dNTP$ would eventually disassociate from the DNA and DNAP complex without the ability to transition past state S_{2B}. Therefore, the system vacillates between states S_1, S_{2A}, and S_{2B}. In the experiment conducted by Lieberman et al., it was also shown that transitions were only between adjacent states and all rates were exponential. In addition, the transition rate from S_{2A} to S_{2B} was proportional to the concentration of $dNTP$ or $r_{23} = k[dNTP]$ [32]. However, for the purpose of this work, the concentration of $dNTP$ was fixed, so r_{23} was treated as a fixed value. Given the exponential transition rates, this system could be treated as a three state continuous time Markov chain as represented in the state diagram in

Figure 4.2A.

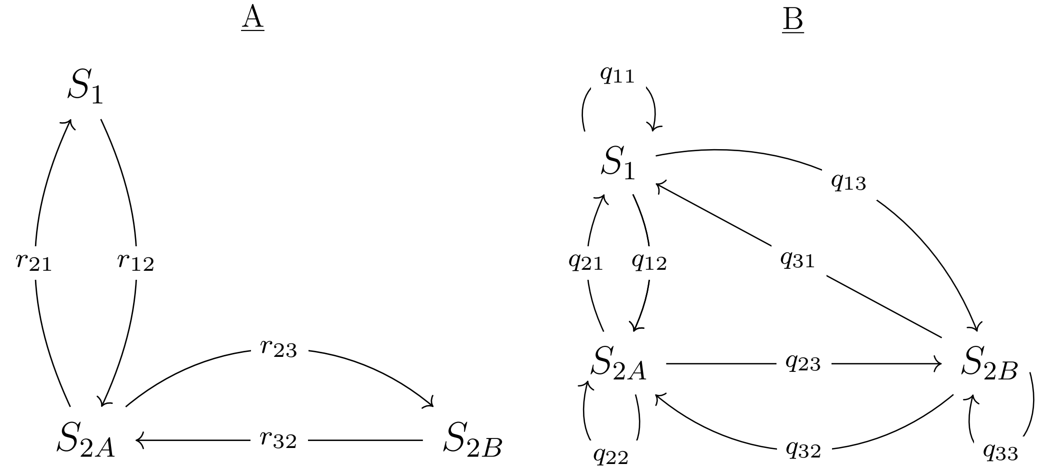


Figure 4.2: State diagrams of the continuous time 3-state system with transition rates r_{ij} (A) and discrete time 3-state system with transition probabilities q_{ij} (B)

The state diagram of the resulting 3-state discrete time system was depicted in Figure 4.2B where q_{ij} is the transition probability from state i to j. Since it was possible for more than one transition within a single timestep (t_i to t_{i+1}), there are transition probabilities that do not exist in Figure 4.2A. Given the continuous time Markov model from Figure 4.2A, the associated discrete system was also Markovian. Furthermore, given the transition probabilities the transition rates can be calculated and vice versa given the equations listed as (4.1) to (4.4). For further explanation of the Markov properties of both the continuous time system and the discrete time system as well as the mapping between the two, see Section 2.4.2.

Recall, that the observation of state was made indirectly by measuring the applied electric current through the nanopore. In the first experiment, as discussed in Chapter 3, two distinct current levels were associated with states S_1 and S_{2A} [31]. For the datasets discussed, the Baum-Welch algorithm, Viterbi training, and a Gibbs sampler were applicable, but the Baum-Welch algorithm and Gibbs

$$\boldsymbol{R} = \begin{bmatrix} -r_{12} & r_{21} & 0 \\ r_{12} & -\left(r_{21} + r_{23}\right) & r_{32} \\ 0 & r_{23} & -r_{32} \end{bmatrix} \qquad \boldsymbol{Q} = \begin{bmatrix} q_{11} & q_{21} & q_{31} \\ q_{12} & q_{22} & q_{32} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{Q} = \sum_{d=0}^{D} \frac{\left(\mathbf{R}(\Delta t)\right)^{d}}{d!} + o(\Delta t^{D}) \tag{4.2}$$

$$\boldsymbol{R} = \boldsymbol{S}_0 - \frac{1}{2}\left(\Delta t\right)\boldsymbol{S}_0^2 + \frac{1}{3}\left(\Delta t\right)^2\boldsymbol{S}_0^3 - \frac{1}{4}\left(\Delta t\right)^3\boldsymbol{S}_0^4 + o\left(\Delta t^3\right) \tag{4.3}$$

$$\boldsymbol{S}_0 = \frac{\boldsymbol{Q} - \boldsymbol{I}}{\Delta t} \tag{4.4}$$

sampler were more advantageous. However, as discussed in Section 1.1, the three state experiment did not have three distinct current levels. Lieberman and her colleagues found that both S_{2A} and S_{2B} produced the same signal [32]. Then the system only had two distinct current levels. Let μ_{S_1} be the signal produced when the system was in state S_1 and let μ_{S_2} be the signal when the system was in state S_{2A} or S_{2B}. From this, the continuous time system and the discrete time system state diagram were pictured with current levels in Figure 4.3.

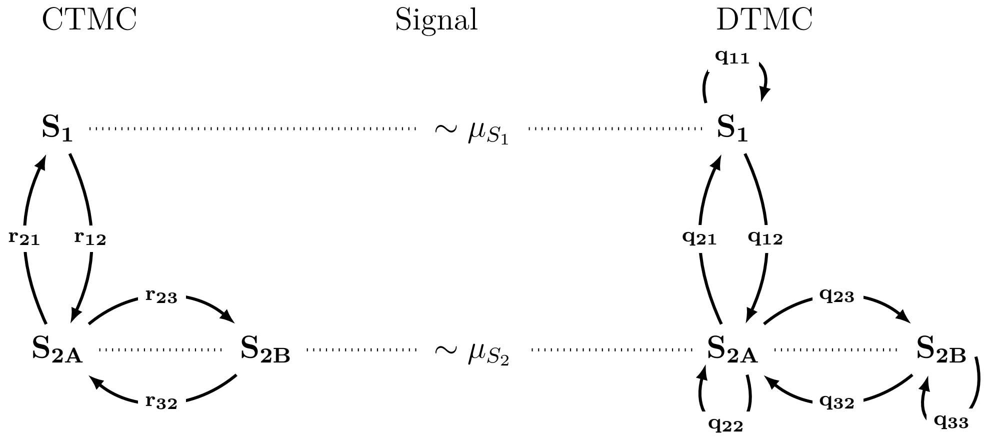


Figure 4.3: The state diagrams of the CTMC and the discrete time markov chain (DTMC) where $q_{13} = q_{31} = 0$ and the signals associated to each state

For the three state system, notation from Section 2.4.2 was used. The current was measured at regularly spaced times $\boldsymbol{t}_1^T = (t_1, t_2, ..., t_T)$ and the i^{th} measurement of current was denoted as y_i. The unobserved state at time t_i was represented by the indicator $\boldsymbol{z}_i = (z_{i1}, z_{i2}, z_{i3})$ defined such that $z_{im} = 1$ if the system was in state m at time t_i, otherwise $z_{im} = 0$. For notational efficiency, let $m = 1$, 2 and 3 represent S_1, S_{2A}, and S_{2B} respectively. The joint likelihood was listed as (4.5) and was the same as the likelihood for the k state system with k distinct signals except for $p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})$. Following that, greater detail on the joint likelihood can be found in Section 2.4.2.

$p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})$ for k states (4.5)

$$\begin{aligned}
p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta}) &= p(\boldsymbol{z}_1|\boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) &= \prod_{j=1}^{k} \left[p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)j} = 1)\right]^{\boldsymbol{z}_{(i-1)j}} \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{(i-1)j} = 1) &= \prod_{l=1}^{k} q_{jl}^{z_{il}} \\
p(\boldsymbol{z}_1|\boldsymbol{\theta}) &= \prod_{j=1}^{k} \rho_j^{z_{1j}} \\
y_i|z_{i1} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_{S_1}, \sigma^2) \\
y_i|z_{i2} = 1 \cup z_{i3} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_{S_2}, \sigma^2) \\
p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) &= \left[p(y_i|z_{i1} = 1, \boldsymbol{\theta})\right]^{z_{i1}} \left[p(y_i|z_{i2} = 1 \cup z_{i3} = 1, \boldsymbol{\theta})\right]^{z_{i2}+z_{i3}}
\end{aligned}$$

As a first test, the Baum-Welch algorithm and the Gibbs sampler from Section 2.4.2 was applied naively to the 3 state system. Since there was much less information than provided in Chapter 3, it was assumed that q_{13} and q_{31} were zero as diagrammed in Figure 4.3 which is approximately true given a sufficiently small timestep (Δt).

4.1.1 Dataset

As with the 2 state system, the algorithms were evaluated against computer generated idealized datasets. Again, the Gillespie algorithm [19] was used to generate the state in continuous time or $X(t)$ where $X(t) = 1$, 2 and 3 denotes the system was in state S_1, S_{2A}, or S_{2B} at time t respectively. $\boldsymbol{z}_0^T$ was taken from $X(t_i)$ for $i = 1$ to T and y_i was generated from $p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})$. As before, the complex was arbitrarily started in state S_1. However, the three state system has two possible transitions from S_{2A}. For this, $test_{21}$ and $test_{23}$ were generated where $test_{21} \sim \exp(r_{21})$ and $test_{23} \sim \exp(r_{23})$. If $test_{21} < test_{23}$, the complex transitioned to state S_1, if $test_{21} > test_{23}$ then the complex transitioned to state S_{2A}. Recall from Chapter 3, E_j is the time of the j^{th} event or transition and ΔE_j be $E_j - E_{j-1}$.

The Gillespie algorithm for the 3-state system

```
1. Set $j = 1$, $t = 0$, and $X(0) = 1$.
2. Generate $E_1 \sim \exp(r_{12})$
3. $t = E_1$, $j = j + 1$, $X(t) = 2$
4. while $t < t_T$
    (a) if $X(t) = 1$
        i. Generate $\Delta E_j \sim \exp(r_{12})$
        ii. $E_j = \Delta E_j + E_{j-1}$, $t = E_j$, and $X(t) = 2$
    (b) if $X(t) = 2$
        i. Generate $test_{21} \sim \exp(r_{21})$
        ii. Generate $test_{23} \sim \exp(r_{23})$
        iii. if $test_{21} < test_{23}$
            A. $\Delta E_j = test_{21}$
            B. $E_j = \Delta E_j + E_{j-1}$, $t = E_j$, and $X(t) = 1$
        iv. if $test_{21} > test_{23}$
            A. $\Delta E_j = test_{23}$
            B. $E_j = \Delta E_j + E_{j-1}$, $t = E_j$, and $X(t) = 3$
    (c) if $X(t) = 3$
        i. Generate $\Delta E_j \sim \exp(r_{32})$
        ii. $E_j = \Delta E_j + E_{j-1}$, $t = E_j$, and $X(t) = 2$
    (d) $j = j + 1$
```

For $\boldsymbol{t}_0^T = (t_0, t_1, ..., t_T)$, $X(t_i)$ was mapped to $\boldsymbol{z}_i$ and y_i was drawn from $y_i \sim p(y_i|\boldsymbol{z}_i, \boldsymbol{\Theta})$

Analysis of performance of traditional hidden Markov modeling

A 10 second dataset was generated with $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma = 3$, and was sampled at 100 kHz. As with the 2 state algorithm, the Baum Welch algorithm was run first. To rule out the effect caused by a poor initial guess, $\hat{\boldsymbol{\theta}}_0$ included the true values of μ_{S_1}, μ_{S_2}, σ, and the approximated values of q_{ij} from 4.2. Given the low amount of information, the algorithm searched in the vicinity of the solution and then failed after 134 iterations. Since there is very little change in the likelihood between being in state S_{2A} and S_{2B}, the algorithm could continually change values for $\mathbb{E}[\boldsymbol{z}_1^T]$ with little to no penalty. A similar experiment was run with the Gibbs sampler, using the same values for the initial guess of θ_0. $15,000$ draws from the posterior of $\boldsymbol{\theta}$ were simulated. The histograms of q_{12}, q_{21}, q_{23}, and q_{32} can be seen in Figure 4.4.

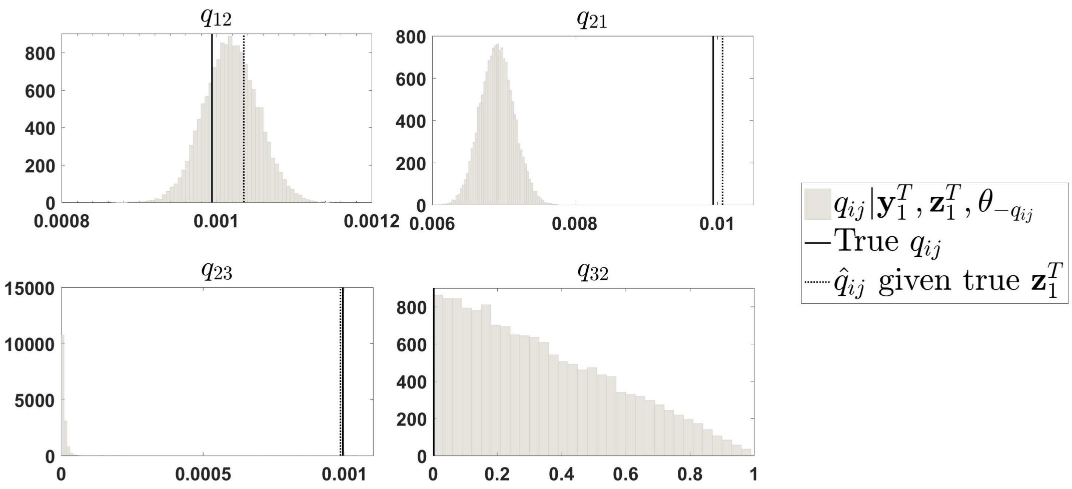


Figure 4.4: The histograms of q_{12}, q_{21}, q_{23}, and q_{32}

From the histograms of posteriors in Figure 4.4, it was easy to see that the posteriors were not good estimated for the values of q_{ij}. Furthermore, the indiscernible signals for state S_{2A} and S_{2B} greatly affected inference on q_{23} in q_{32}. The values of q_{23} were near 0, and q_{32} seemed to explore the entire space from 0 to 1. Since the value of q_{23} was close to zero, the posterior of $\boldsymbol{z}_1^T$ was rarely in state S_{2B}. Therefore, the value of q_{21} was underestimated since the stays in S_{2B} were incorrectly attributed to S_{2A}. Using true values for both the initial value for the Baum-Welch algorithm ($\hat{\boldsymbol{\theta}}_0$) and the Gibbs sampler ($\boldsymbol{\theta}_0$) produced undesirable results, and Viterbi Training was shown to be less accurate. Given three states with only two discernible signals, the traditional methodologies for discrete state systems were insufficient. Therefore, a composite state that included states S_{2A} and S_{2B} was considered.

4.1.2 The composite state

The three state system with only two distinct signals introduced added complexity to inference. Since the state was not directly measurable and there were only two discernible signals, changing from $z_{i2} = 1$ to $z_{i3} = 1$ or vice versa could yield very little change in the value of the likelihood. This made finding the MLE using traditional hidden Markov modeling methods very difficult and was confirmed empirically by the results in Section 4.1.1. Therefore, the problem was reconsidered using only the "visible" states. The first of which was the pre-translocation state, or state S_1. The second, referred to as the composite state denoted as S_2, was the union of states S_{2A} and S_{2B} ($S_2 = S_{2A} \cup S_{2B}$). The two "visible" states two were illustrated in Figure 4.5 where $q_{*1}(d_h)$ is the transition probability from S_2.

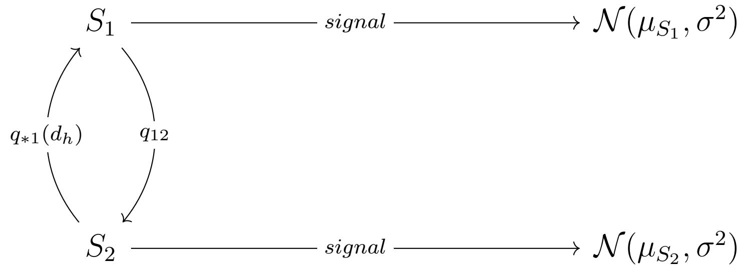

S_1
signal
$\mathcal{N}(\mu_{S_1}, \sigma^2)$
$q_{*1}(d_h)$
q_{12}
S_2
signal
$\mathcal{N}(\mu_{S_2}, \sigma^2)$

Figure 4.5: The visible portion of the discrete three state model

q_{12} is the same as depicted in Figure 4.3 and therefore is memoryless. However, the escape probability from the composite state (S_2) was not Markovian. By assuming only the transitions in Figure 4.3 were possible, the escape probability from the composite state was a mixture of geometric distributions. Furthermore, the transition probabilities in Figure 4.3 can be written in terms of the parameters of the escape probability from S_2. The escape probability from S_2 was written in (4.6) and denoted as $q_{*1}(d_h)$ where d_h is the number of consecutive observations in the h^{th} stay in S_2. Following that, q_{21}, q_{23}, and q_{32} were written in terms of the escape probability from S_2 in 4.7-4.9.

$$q_{*1}(d_h) = w(1-\varphi_1)^{d_h-1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h-1}\varphi_2 \tag{4.6}$$

$$q_{21} = \varphi_2 + w(\varphi_1 - \varphi_2) \tag{4.7}$$

$$q_{23} = \frac{(\varphi_1-\varphi_2)^2 - [(\varphi_1-\varphi_2) - 2w(\varphi_1-\varphi_2)]^2}{2[\varphi_2 + w(\varphi_1-\varphi_2)]} \tag{4.8}$$

$$q_{32} = \varphi_2 - w(\varphi_1-\varphi_2) - \frac{(\varphi_1-\varphi_2)^2 - [(\varphi_1-\varphi_2) - 2w(\varphi_1-\varphi_2)]^2}{2[\varphi_2 + w(\varphi_1-\varphi_2)]} \tag{4.9}$$

Given q_{12} and (4.7)-(4.9), inference on $\boldsymbol{z}_1^T = (\boldsymbol{z}_1, \boldsymbol{z}_1, ..., \boldsymbol{z}_T)$ could be made using the previously discussed hidden Markov modeling algorithms. Although there was insufficient information for convergence on $\boldsymbol{z}_1^T$, there was sufficient information to converge on the values for z_{i1} and $z_{i2} + z_{i3}$. In turn, this was enough information for inference on q_{12}, μ_{S_1}, μ_{S_2}, and σ^2. Furthermore, the values of z_{i1} and $z_{i2} + z_{i3}$, were used to to generate $\boldsymbol{d} = (d_1, d_2, ..., d_h, ..., d_n)$ where d_h is the consecutive observations in the composite state S_2 for the h^{th} stay in S_2. From that, inference on φ_1, φ_2, and w was made.

Consecutive observations in the composite state

To understand the consecutive observations in the composite state and consequently the transition probability from the composite state, an arbitrary sequence of consecutive observations in state S_2 was studied. Without loss of generality, let $\boldsymbol{z}_{i+1}$ be the first observation in the composite state, Then $\boldsymbol{p}(\boldsymbol{z}_{i+2}|\boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ was studied where states S_1, S_{2A}, and S_{2B} were considered. Since all three states were considered, as previously discussed, the probabilities of transitioning between the states, were Markovian. Furthermore, $\boldsymbol{p}(\boldsymbol{z}_{i+2}|\boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ could be expressed as a single matrix and was listed as (4.11) to (4.13). To reduce the number of variables, q_{11} was written as $1 - q_{12}$, q_{22} was written as $1 - q_{21} - q_{23}$, and q_{33} was written as $1 - q_{32}$

$$\boldsymbol{p}(\boldsymbol{z}_{i+2}|\boldsymbol{z}_{i+1}, \boldsymbol{\Theta}) = \boldsymbol{Q}\boldsymbol{z}_i' \tag{4.10}$$

$$\boldsymbol{p}(\boldsymbol{z}_{i+2}|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) = \begin{bmatrix} p(\boldsymbol{z}_{(i+2)1} = 1|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ p(\boldsymbol{z}_{(i+2)2} = 1|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ p(\boldsymbol{z}_{(i+2)3} = 1|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \end{bmatrix} \tag{4.11}$$

$$\boldsymbol{z}_{i+1} = \begin{bmatrix} z_{(i+1)1} & z_{(i+1)2} & z_{(i+1)3} \end{bmatrix} \tag{4.12}$$

$$\boldsymbol{Q} = \begin{bmatrix} 1 - q_{12} & q_{21} & 0 \\ q_{12} & 1 - (q_{21} + q_{23}) & q_{32} \\ 0 & q_{23} & 1 - q_{32} \end{bmatrix} \tag{4.13}$$

The matrix above, $\boldsymbol{Q}$, has some extra information for the aforementioned purposes. First, since the dwell time in S_2 was being studied, then it was assumed that $z_{(i-1)1} = 0$. Therefore, the first column was not needed. Second, $q_{*1}(d_h)$ was derived by computing the probability of staying in S_2, so it was not necessary to compute the values in the first row. Therefore, the probability of staying in state S_2 could be described using the reduced matrices in (4.14).

$$\boldsymbol{Q}_{S_2} = \begin{bmatrix} 1 - q_{21} - q_{23} & q_{32} \\ q_{23} & 1 - q_{32} \end{bmatrix}$$

$$\begin{bmatrix} p(\boldsymbol{z}_{(i+2)2} = 1|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ p(\boldsymbol{z}_{(i+2)3} = 1|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \end{bmatrix} = \boldsymbol{Q}_{S_2} \begin{bmatrix} z_{(i+1)2} \\ z_{(i+1)3} \end{bmatrix} \tag{4.14}$$

$$\begin{bmatrix} p(\boldsymbol{z}_{(i+n)2} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ p(\boldsymbol{z}_{(i+n)3} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2|\boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \end{bmatrix} = \boldsymbol{Q}_{S_2}^{n-1} \begin{bmatrix} z_{(i+1)2} \\ z_{(i+1)3} \end{bmatrix} \tag{4.15}$$

(4.14) computes the probability of remaining in S_2 at time t_{i+1} to t_{i+2}. However, it was necessary to derive the probability of a general set of consecutive observations remaining in S_2 from time t_{i+1} to t_{i+n}. For this, two quantities were computed, $p(\boldsymbol{z}_{(i+n)2} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2|\boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ and $p(\boldsymbol{z}_{(i+n)3} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2|\boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ where $\boldsymbol{z}_{i+1}^{i+n-1} \in S_2$ denotes that the system was observed in the composite state from $i+2$ to $i+n-1$. Since the three state system was Markovian, this was done by multiplying $\boldsymbol{Q}_{S_2}$ $(n-1)$ consecutive times as seen in (4.15). Furthermore, $\boldsymbol{Q}_{S_2}$ was diagonalizable. Therefore, a diagonal matrix $\mathbf{D}$ and matrix $\mathbf{M}$ exist such that $\boldsymbol{Q}_{S_2}^{n-1} = \mathbf{M}\mathbf{D}^{n-1}\mathbf{M}^{-1}$. Expressing $\mathbf{D}$ as (4.16), $\mathbf{M}\mathbf{D}^{n-1}\mathbf{M}^{-1}$ was simplified in (4.17) where each l_i was the resulting constant from multiplying $\boldsymbol{D}$ by two general unknown matrices ($\boldsymbol{M}$ and $\boldsymbol{M}^{-1}$). Since $\boldsymbol{z}_{i+1}$ was the first observation in the composite state and the assumption $q_{13} = 0$ was used then $z_{(i+1)2} = 1$ and $z_{(i+1)3} = 0$ resulting in (4.18). (4.18) gives the probability $z_{(i+n)2} = 1$ and $z_{(i+n)3} = 1$ separately, but the interest was the probability of remaining in S_2 which was denoted as $p(\boldsymbol{z}_{i+2}^{i+n} \in S_2|\boldsymbol{z}_{i+1}, \boldsymbol{\theta})$. This result was listed as (4.19).

$$\boldsymbol{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{4.16}$$

$$\mathbf{M}\mathbf{D}^{n-1}\mathbf{M}^{-1} = \begin{bmatrix} l_1\lambda_1^{n-1} + l_2\lambda_2^{n-1} & l_3\lambda_1^{n-1} + l_4\lambda_2^{n-1} \\ l_5\lambda_1^{n-1} + l_6\lambda_2^{n-1} & l_7\lambda_1^{n-1} + l_8\lambda_2^{n-1} \end{bmatrix} \tag{4.17}$$

$$\begin{bmatrix} p(\boldsymbol{z}_{(i+n)2} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ p(\boldsymbol{z}_{(i+n)3} = 1, \boldsymbol{z}_{i+2}^{i+n-1} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} l_1 \lambda_1^{n-1} + l_2 \lambda_2^{n-1} \\ l_5 \lambda_1^{n-1} + l_6 \lambda_2^{n-1} \end{bmatrix} \tag{4.18}$$

$$p(\boldsymbol{z}_{i+2}^{i+n} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) = (l_1 + l_5)\lambda_1^{n-1} + (l_2 + l_6)\lambda_2^{n-1} \tag{4.19}$$

Finally, (4.19) could be used to calculate the transition probability given n consecutive observations in the composite state. Since, (4.19) was $p(\boldsymbol{z}_{i+2}^{i+n} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta})$, it was also the probability that there were at least n consecutive observations. Following that, $p(\boldsymbol{z}_{i+2}^{i+n+1} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ gives the probability that there were at least $n+1$ consecutive observations. The transition out of the composite state after exactly n consecutive observation in S_2 or $q_{*1}(n)$ was $p(\boldsymbol{z}_{i+2}^{i+n} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) - p(\boldsymbol{z}_{i+2}^{i+n+1} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta})$ which resulted in an alternate representation of a mixture of geometric distribution listed as 4.20.

$$\begin{aligned} q_{*1}(n) &= p(\boldsymbol{z}_{i+2}^{i+n} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) - p(\boldsymbol{z}_{i+2}^{i+n+1} \in S_2 | \boldsymbol{z}_{i+1}, \boldsymbol{\theta}) \\ q_{*1}(n) &= (l_1 + l_5)\lambda_1^{n-1} + (l_2 + l_6)\lambda_2^{n-1} - [(l_1 + l_5)\lambda_1^{n} + (l_2 + l_6)\lambda_2^{n}] \\ q_{*1}(n) &= (l_1 + l_5)\lambda_1^{n-1}(1 - \lambda_1) + (l_2 + l_6)\lambda_2^{n-1}(1 - \lambda_2) \end{aligned} \tag{4.20}$$

The representation (4.20) was equivalent to the more traditional for of the mixture model in (4.6) where $\lambda_i = 1 - \varphi_i$. Furthermore, since this is a distribution, $l_1 + l_5 + l_2 + l_6 = 1$ so w was used where $w = l_1 + l_5$ and $1 - w = l_2 + l_6$. Given the definitions of (4.16) and (4.17), each λ_i was an eigenvalue of $\boldsymbol{Q}_{S_2}$. This made it possible to write each λ_i and φ_i it terms of the transition probabilities of the hidden Markov model, which were listed as (4.21)-(4.24).

$$\lambda_1 = \frac{2 - q_{21} - q_{23} - q_{32} - k}{2} \tag{4.21}$$

$$\lambda_2 = \frac{2 - q_{21} - q_{23} - q_{32} + k}{2} \tag{4.22}$$

$$k = \sqrt{2q_{21}q_{23} - 2q_{21}q_{32} + 2q_{23}q_{32} + q_{21}^2 + q_{23}^2 + q_{32}^2}$$

$$\varphi_1 = \frac{q_{21} + q_{23} + q_{32} + k}{2} \tag{4.23}$$

$$\varphi_2 = \frac{q_{21} + q_{23} + q_{32} - k}{2} \tag{4.24}$$

Finally, w was written in terms of the transition probabilities of the hidden Markov model. For this, the assumption $q_{13} = 0$ was used again. If $\boldsymbol{z}_{(i+1)2} = 1$, then the probability of transition out of S_2 on the first timestep was q_{21}. Therefore, to calculate w in terms of the transition probabilities of the hidden Markov model, $q_{*1}(1)$ was set equal to q_{21}. This resulted in (4.25).

$$\begin{aligned}
q_{*1}(1) &= q_{12} \\
q_{12} &= w(1-\varphi_1)^{1-1}\varphi_1 + (1-w)(1-\varphi_2)^{1-1}\varphi_2 \\
q_{12} &= w\varphi_1 + (1-w)\varphi_2 \\
q_{12} &= w\left(\frac{q_{21} + q_{23} + q_{32} + k}{2}\right) + (1-w)\left(\frac{q_{21} + q_{23} + q_{32} - k}{2}\right) \\
q_{12} &= w(k) + \left(\frac{q_{21} + q_{23} + q_{32} - k}{2}\right) \\
w &= \frac{q_{21} - q_{23} - q_{32} + k}{2k} \qquad (4.25)
\end{aligned}$$

From this, algorithms were built to make inference on r_{ij}. This will be discussed in the following sections.

4.2 Inference on a CTMC with 3 states and 2 distinct signals

In Chapter 3, inference on a 2 state system with 2 distinct signals was discussed. To calculate the MLE of the transition rates the Baum-Welch algorithm and Viterbi training were applied. For the datasets explored, the Baum-Welch was the preferred method of inference due to higher quality inference than Viterbi training while having reasonable compute times. Unfortunately, the Baum-Welch Algorithm is a special case of the Expectation Maximization Method [1] that iteratively computes $\mathbb{E}[\boldsymbol{z}_1^T]$ and then maximizes the parameters based on $\mathbb{E}[\boldsymbol{z}_1^T]$ [4]. The interpretation of $\mathbb{E}[\boldsymbol{z}_1^T]$ to consecutive observations in state S_2 (d_h) was not straightforward, making inference on (4.6) difficult. Therefore, a method that makes inference on $\boldsymbol{z}_1^T$ was preferential to one that makes inference on $\mathbb{E}[\boldsymbol{z}_1^T]$. The other approach called the Viterbi training or segmental K-means algorithm [24] [44] discussed in Chapter 3 made inference on the most likely state sequence of $\boldsymbol{z}_1^T$ given the parameters. This path (Viterbi path) can be found employing the Viterbi algorithm [58]. Since the Viterbi path provides a single sequence for $\boldsymbol{z}_1^T$, values of the consecutive observations in the composite state (d_h) were straight forward to calculate. Therefore, the solutions to the Viterbi path could be used to compute all parameters of interest in an iterative method similar to the EM. Viterbi training is known to be less computationally intensive than Baum-Welch, but Viterbi training is not asymptotically unbiased [42]. The asymptotically biased estimates of the algorithm were found to have an impact, and those will be discussed later.

Chapter 3 also discussed the calculation of the posteriors of the parameters using a Gibbs sampler [53] [9]. This used a variation on the forward filtering backward sampling algorithm to make inference on $\boldsymbol{z}_0^T$ and the parameters. A weakness of the aforementioned algorithm is long computation times. The computation time is particularly limiting because the experiments above require large datasets to make inference with acceptable accuracy.

4.2.1 Limitations

The goal of this research was to make inference on the transition rates of a three-state system when only there were only two distinct signals. Without a sufficient number of observations and favorable conditions, inference was difficult or of poor quality. Unfortunately, there was not a set range of transition rates that performed poorly. Instead, problems occurred on a sliding range that can be alleviated through more observations or an adjustment of Δt. Following that, this section explores the possible types of problems that can occur through the lens of the problem introduced in Section 4.1. Since it was observed by Liebermen and her colleagues [31] [32] that r_{21} was much larger than r_{23} and r_{32}, for this section r_{21} was arbitrarily set to 1000 and r_{23} and r_{32} were varied from 50 to 500. Furthermore, the sampling rate was set to $\Delta t = \frac{1}{100000}$, as used with the original experiment.

The first limitation comes from the assumptions necessary to derive the probability of transition from the composite state or state S_2. For $q_{*1}(d_h)$, it was assumed that $q_{13} \approx q_{13} \approx 0$. Occurrences where the system changes from state S_1 to S_{2B} and from S_{2B} to S_1 were not considered in $q_{*1}(d_h)$. A good tool to study the probability of q_{13} and q_{31} was the second order approximations of q_{ij}. For this, the second order approximations in terms of Δt for q_{12} and q_{13} were listed as (4.26) and (4.27) respectively. Since the algorithm assumes q_{13} does not account for any of the transitions to the composite state, it was important that q_{13} was very small relative to q_{12}. If q_{13} was small relative to q_{12}, most of the transitions into state S_2 were from state S_1 to S_{2A} as assumed. By examining (4.26) and (4.27), it can be seen that q_{12} contains a first degree term but q_{13} does not. Furthermore, the second degree of q_{12} and q_{13} both have negative effects on the desired ratio of q_{12} to q_{13}. The same can be observed in the cases of q_{21} and q_{31}. Therefore, the ratios of $q_{13} : q_{12}$ and $q_{31} : q_{21}$ can be regulated by choosing Δt small enough such that the second (and all following degrees) were made negligible, leaving only the desired first degree transitions.

$$q_{12} = r_{12}\Delta t - (r_{12}^2 + r_{12}r_{21} + r_{12}r_{23})\frac{\Delta t^2}{2!} + o(\Delta t^2) \tag{4.26}$$

$$q_{13} = r_{12}r_{23}\frac{\Delta t^2}{2!} + o(\Delta t^2) \tag{4.27}$$

Another factor that contributes to the quality of inference were the values φ_1 to φ_2. If the values of φ_1 and φ_2 were too close or if φ_2 was too small, inference on φ_1, φ_2, and w was difficult. Recall from (4.23) and (4.24) that φ_1 and φ_2 differ by k which was expressed differently in (4.28). Furthermore, k was expressed in terms of r_{ij} in equation (4.29) using the first degree expansions of q_{ij}. From this, k was approximately related to the difference of $(r_{21} + r_{23} + r_{32})^2(\Delta t) - 4r_{21}r_{32}$. In the case that the difference was too small, Δt can be adjusted. However, if $4r_{21}r_{32}\Delta t$ was too close to 0, then $k \approx q_{21} + q_{23} + q_{32}$ and φ_2 becomes very close to 0. This can be alleviated by making Δt larger. Plots comparing $\frac{\varphi_1}{\varphi_2}$ and φ_2 to values of r_{23} and r_{32} were placed in Figure 4.6 when r_{21} was held constant.

$$k = \sqrt{(q_{21} + q_{23} + q_{32})^2 - 4q_{21}q_{32}} \tag{4.28}$$

$$k \approx \sqrt{\left[(r_{21} + r_{23} + r_{32})^2(\Delta t) - 4r_{21}r_{32}\right](\Delta t)} \tag{4.29}$$

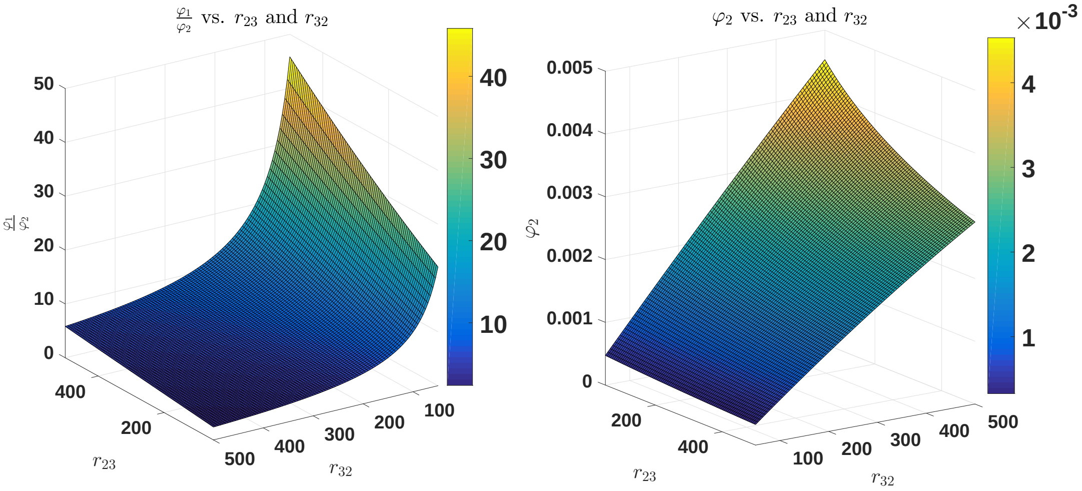


Figure 4.6: Comparing $\frac{\varphi_1}{\varphi_2}$ to r_{23} and r_{32} when $r_{21} = 1000$ and $\Delta t = \frac{1}{100000}$

Since r_{21} was held constant in Figure 4.6, as noted in equation (4.28), r_{32} had the largest effect on both quantities. Therefore, Δt needed to be chosen such that $4r_{21}r_{32}$ was smaller than $(r_{21}+r_{23}+r_{32})^2(\Delta t)$ without $4r_{21}r_{32}(\Delta t)$ being too close to zero.

The last factor considered for the composite state was w. If w was too close to 1, inference was difficult because there were very few observations of φ_2. If w was too close to 0, inference was difficult because there were very few observations of φ_1. Unfortunately, w did not lend itself to simplification in the same manner as k, but the value of w was plotted against r_{23} and r_{32} in Figure 4.7. As before, r_{21} was set to 1000 and $\Delta t = \frac{1}{100000}$. Since r_{21} was at least twice r_{23} and r_{32} in the example presented, w was never close to 0. However, it is possible this could be problematic in the general application of these methods. As seen in Figure 4.7, w does approach one when both r_{23} and r_{32} were too small.

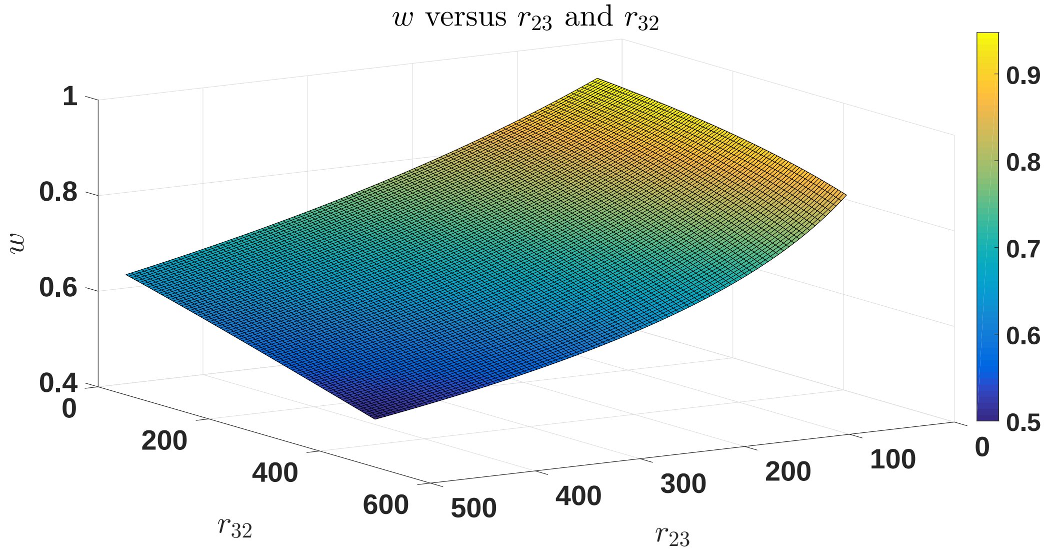


Figure 4.7: Comparing w to r_{23} and r_{32} when $r_{21} = 1000$ and $\Delta t = \frac{1}{100000}$

It would be convenient to list values of q_{ij} where these methods became ineffective. Unfortunately, these values are not fixed. Acceptable, values of $\frac{\varphi_1}{\varphi_2}$, φ_2, and w depend on the values of q_{ij} and the amount of information available. These, in turn, rely on the values of Δt and number of data points. It could be the case that problems could be resolved by changing Δt. If Δt was already optimal, problems could be resolved with additional observations. Therefore, assuming an optimal Δt, bad q_{ij} could only be defined for fixed a number of data points or compute time. Following this, Figures 4.6 and 4.7 were helpful in the case that the algorithm performed poorly.

Unfortunately, processing more data was not free. For reference, the training algorithm for point estimates took from 4-10 minutes to yield a biased estimator for a dataset with 1,000,000 data points. The corresponding Gibbs sampler took just over a week of compute time.

4.2.2 A parameter training algorithm based on Viterbi training

It was shown that traditional HMM methods were ineffective for making inference on a three state system with two distinct signals. Therefore, a composite state was considered which was explained in detail in Section 4.1.2. Given the new composite state S_2, inference was made on $\boldsymbol{\theta} = (\boldsymbol{\rho}, q_{12}, \varphi_1, \varphi_2, w, \mu_{S_1}, \mu_{S_2}, \sigma^2)$. Similar to hidden Markov modeling, inference was made on the hidden state $\boldsymbol{z}_1^T$ which allowed inference on $\boldsymbol{\theta}$. This resulted in an iterative algorithm that in this research exhibited similar behaviors to Viterbi training. Following the same notation as used previously, let $\hat{a}_k$ be the estimate of a after k iterations, $(\hat{a}_i)_k$ be the same if a has a subscript, and $v_k\left(\boldsymbol{z}_1^T\right)$ is the Viterbi path after k iterations. Since the d_h, or the number of consecutive observations in state S_2 were taken from the Viterbi path, the estimate of d_h from the Viterbi path was denoted $v_k\left(d_h\right)$. Like the Viterbi training algorithm, this algorithm requires an initial guess $\left(\hat{\boldsymbol{\theta}}_0\right)$ and since the algorithm is a local optimizer it is important that $\hat{\boldsymbol{\theta}}_0$ is close enough to the true $\boldsymbol{\theta}$ that the local optimum found was also the global optimum. To assess convergence, $\left|\left|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k-1}\right|\right|_\infty$ was measured after each iteration (k) where $\left|\left|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k-1}\right|\right|_\infty$ denotes the infinity norm. When $\left|\left|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k-1}\right|\right|_\infty < \delta$ where δ was an arbitrary small number, it was concluded that the algorithm had converged and the algorithm was stopped. Therefore, the algorithm runs within a while loop and to initialize the while loop when $k = 0$, $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty$ was set equal to an arbitrary number greater than δ.

A Parameter training algorithm for 3 states with 2 distinct signals (4.30)

```
I. Pick $\hat{\boldsymbol{\theta}}_0$
II. Set $k = 0$ and define $\left|\left|\hat{\boldsymbol{\theta}}_0 - \hat{\boldsymbol{\theta}}_{-1}\right|\right|_\infty > \delta$
```

```
III. While $\left|\left|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k-1}\right|\right|_\infty > \delta$
        (i) k=k+1
       (ii) Compute $v_k\left(\boldsymbol{z}_1^T\right)$ from (4.41)
      (iii) Compute $\hat{\boldsymbol{\rho}}_k$ from (4.42)
       (iv) Compute $\left(\hat{q_{12}}\right)_k$ from (4.43)
        (v) Compute $\left(\hat{\mu_{S_1}}\right)_k$ from (4.44)
       (vi) Compute $\left(\hat{\mu_{S_2}}\right)_k$ from (4.45)
      (vii) Compute $\hat{\sigma^2}_k$ from (4.46)
     (viii) Harvest $v_k\left(\boldsymbol{d}\right)$ from $v_k\left(\boldsymbol{z}_1^T\right)$
       (ix) Compute $\left(\varphi_1, \varphi_2, \varphi_2\right)_k$ from (4.50)
```

The likelihood functions

Following the discussion above, the system was described using two likelihood functions. The first of which was listed as (4.31)-(4.35). This was the traditional Hidden Markov Model where it was only important to converge on $v_k\left(z_{i1}\right)$ and $v_k\left(z_{i2}\right) + v_k\left(z_{i3}\right)$ for all i. Recall, $q_{21}(\boldsymbol{\theta})$, $q_{23}(\boldsymbol{\theta})$, and $q_{23}(\boldsymbol{\theta})$ are functions of $\boldsymbol{\theta}$ and were listed as (4.7)-(4.9).

$$p(\boldsymbol{y}_0^T, \boldsymbol{z}_0^T|\boldsymbol{\theta}) = p(\boldsymbol{z}_0|\boldsymbol{\theta}) \left[\prod_{i=1}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}, \boldsymbol{\theta})\right] \left[\prod_{i=0}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})\right] \tag{4.31}$$

$$p(\boldsymbol{z}_0|\boldsymbol{\theta}) = \rho_1^{z_{10}} \rho_2^{z_{20}} \rho_3^{z_{30}} \tag{4.32}$$

$$p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}, \boldsymbol{\theta}) = \boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{z'_{i-1}} \tag{4.33}$$

$$p(y_i|\boldsymbol{z}_i,\boldsymbol{\theta}) = \left[\mathcal{N}_{y_i}(\mu_{S_1},\sigma^2)\right]^{z_{1i}} \left[\mathcal{N}_{y_i}(\mu_{S_2},\sigma^2)\right]^{z_{2i}+z_{3i}} \tag{4.34}$$

$$\boldsymbol{Q(\theta)} = \begin{bmatrix} 1-q_{12} & q_{21} & 0 \\ q_{12}(\boldsymbol{\theta}) & 1-(q_{21}(\boldsymbol{\theta})+q_{23}(\boldsymbol{\theta})) & q_{32}(\boldsymbol{\theta}) \\ 0 & q_{23}(\boldsymbol{\theta}) & 1-q_{32}(\boldsymbol{\theta}) \end{bmatrix} \tag{4.35}$$

The second was the likelihood of the consecutive observations in state S_2 (or probability of transition from the composite state S_2 after d_h consecutive observations) for n stays in the composite state which was listed as (4.36).

$$p(\boldsymbol{d}|\boldsymbol{\theta}) = \prod_{h=1}^{n} \left[w(1-\varphi_1)^{d_h-1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right] \tag{4.36}$$

The training iteratively maximizes the state sequence $\left(v_k\left(\boldsymbol{z}_1^T\right)\right)$ given $\hat{\boldsymbol{\theta}}_{k-1}$ then $\hat{\boldsymbol{\theta}}_k$ given $v_k\left(\boldsymbol{z}_1^T\right)$. In this case, the parameter training for this project iteratively performs the maximizations listed in (4.37)-(4.39) where $\{\boldsymbol{\rho}_k\ , (\hat{q_{12}})_k\, , (\hat{\mu_{S_1}})_k\ , (\hat{\mu_{S_2}})_k\, , \hat{\sigma^2}_k\Big\}$ and $\{\hat{w}_k, (\hat{\varphi}_1)_k\, , (\hat{\varphi}_2)_k\}$ together are $\hat{\boldsymbol{\theta}}_k$.

$$v_k\left(\boldsymbol{z}_1^T\right) = \underset{\boldsymbol{z_0^T}}{\operatorname{argmax}}\, p\left(\boldsymbol{y}_0^T, \boldsymbol{z}_0^T \left|\hat{\boldsymbol{\theta}}_{k-1}\right.\right) \tag{4.37}$$

$$\left\{\boldsymbol{\rho}_k, (\hat{q_{12}})_k\, , (\hat{\mu_{S_1}})_k\, , (\hat{\mu_{S_2}})_k\, , \hat{\sigma^2}_k\right\} = \underset{q_{12},\mu_{S_1},\mu_{S_2},\sigma^2}{\operatorname{argmax}}\; p\left(\boldsymbol{y}_0^T, v_k\left(\boldsymbol{z}_1^T\right)|\boldsymbol{\theta}\right) \tag{4.38}$$

$$\{\hat{w}_k, (\hat{\varphi}_1)_k\, , (\hat{\varphi}_2)_k\} = \underset{\varphi_1,\varphi_2,w}{\operatorname{argmax}}\, p\left(v_k\left(\boldsymbol{d}\right)|\boldsymbol{\theta}\,\right) \tag{4.39}$$

Calculating $v_k\left(\boldsymbol{z}_1^T\right)|\hat{\boldsymbol{\theta}}_{k-1}$

As stated previously, the most likely state sequence given $\hat{\boldsymbol{\theta}}_{k-1}$ was computed using the Viterbi algorithm. For this, a few intermediate calculations were required at each time step. The first, was the probability the most likely state sequence from $t = t_0$ to t_i ended such that $z_{ri} = 1$ (denoted as $\psi_i(r)$). The second expressed as $\Psi_i(r)$, was the most likely previous state given the sequence from $t = t_0$ to t_i ended such that $z_{ri} = 1$. These two were calculated sequentially from $i = 1$ to $i = T$ as outlined in (4.40).

Calculating $\psi_i(r)$ and $\Psi_i(r)$ (4.40)

```
I. Initialization (The first timestep or $t_1$)
    i. For $m = 1 : k$
        A. $\psi_1(m) = p\left(z_{1m} = 1 \middle| \hat{\theta}_{j-1}\right) p\left(y_1 \mid z_{1m} = 1, \hat{\theta}_{j-1}\right)$
    ii. end
II. For $i = 2 : T$
    i. For $n = 1 : k$
        A. $\psi_i(n) = \left[\max_m \psi_{i-1}(m) \left(\hat{q}_{mn}\right)_{j-1}\right] p\left(y_i | z_{in} = 1, \hat{\theta}_{j-1}\right)$
        B. $\Psi_i(n) = \underset{m}{\operatorname{argmax}}\, \psi_{i-1}(m) \left(\hat{q}_{mn}\right)_{j-1}$
    ii. end
III. end
```

From this, the Viterbi Path was traced backward using (4.41) from $t = t_T$ to $t = t_1$ where m and n were temporary variables for each step of the backward iteration.

Calculating $\left(v_k\left(\boldsymbol{z}_1^T\right)\right)$ (4.41)

```
I. $m = \underset{r}{\operatorname{argmax}}\, \psi_T(r)$
II. $v\left(z_{Tm}\right) = 1$ and $v\left(z_{Tr}\right) = 0 \;\forall\; r \neq m \in \{1, 2, .., k\}$
```

```
III. $n = m$

IV. For $i = T - 1 : 1$

    i. $m = \Psi_{i+1}(n)$

    ii. $v\left(z_{im}\right) = 1$ and $v\left(z_{ir}\right) = 0 \;\forall\; r \neq m \in \{1, 2, .., k\}$

    iii. $v\left(\boldsymbol{z}_i^T\right) = \begin{bmatrix} v\left(\boldsymbol{z}_i\right) \\ v\left(\boldsymbol{z}_{i+1}^T\right) \end{bmatrix}$

    iv. $n = m$

V. end
```

All of the calculations for (4.40) and (4.41) above were listed in terms of probability functions. However, given a time series of sufficient length for inference on $\hat{\boldsymbol{\theta}}$, it was necessary to compute log probabilities to avoid complications caused by underflow. The Viterbi Algorithm was discussed in more detail in Section 2.4.2.

Calculating $\left\{\boldsymbol{\rho}_k, (\hat{q_{12}})_k\,, (\hat{\mu_{S_1}})_k\,, (\hat{\mu_{S_2}})_k\,, \hat{\sigma^2}_k\right\} \middle| v_k\left(\boldsymbol{z}_1^T\right)$

$\boldsymbol{\rho}_k$, $(\hat{q_{12}})_k$, $(\hat{\mu_{S_1}})_k$,$(\hat{\mu_{S_2}})_k$, and $\hat{\sigma^2}_k$ were computed using the likelihood functions and were discussed in more detail in Section 2.4.2. The maximums of $\boldsymbol{\rho}_k$, $(\hat{q_{12}})_k$, $(\hat{\mu_{S_1}})_k$,$(\hat{\mu_{S_2}})_k$, and $\hat{\sigma^2}_k$ were given as (4.43) to (4.46).

$$\hat{\boldsymbol{\rho}}_k = v_k\left(\boldsymbol{z}_1\right) \tag{4.42}$$

$$(\hat{q}_{12})_k = \frac{\sum\limits_{i=1}^{T} v_k\left(\hat{z}_{1(i-1)}\right) v_k\left(z_{2i}\right)}{\sum\limits_{i=1}^{T} v_k\left(z_{1i}\right)} \tag{4.43}$$

$$\left(\hat{\mu}_{S_1}\right)_k = \frac{\sum_{i=1}^{T} v_k\left(z_{1i}\right) y_i}{\sum_{i=1}^{T} v_k\left(z_{1i}\right)} \tag{4.44}$$

$$\left(\hat{\mu}_{S_2}\right)_k = \frac{\sum_{i=1}^{T} \left(v_k\left(z_{2i}\right)_k + v_k\left(z_{3i}\right)_k\right) y_i}{\sum_{i=1}^{T} \left(v_k\left(z_{2i}\right) + v_k\left(z_{3i}\right)\right)} \tag{4.45}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{T} \left[v_k\left(\hat{z}_{1i}\right)\left(y_i - \left(\mu_{S_1}\right)_k\right)^2 + \left(v_k\left(z_{2i}\right) + v_k\left(z_{3i}\right)\right)\left(y_i - \left(\mu_{S_2}\right)_k\right)^2\right]}{T} \tag{4.46}$$

Calculating $\left(\hat{\varphi}_1\right)_k, \left(\hat{\varphi}_2\right)_k, \hat{w}_k | v_k\left(\boldsymbol{d}\right)$

$v_k\left(\boldsymbol{z}_0^T\right)$ determines the values for $v_k\left(\boldsymbol{d}\right)$. Unfortunately, (4.39) can not be calculated immediately as was done with (4.38). Therefore, because it is computationally advantageous, an alternate form, as seen in (4.47), was used.

$$p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w) = \prod_{h=1}^{n} \left[w(1-\varphi_1)^{d_h - 1}\varphi_1\right]^{v_h} \left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h} \tag{4.47}$$

In (4.47), the additional variable v_h indicates which geometric distribution d_h was drawn from. Using the form listed in (4.47), $\left(\hat{\varphi}_1\right)_k$, $\left(\hat{\varphi}_2\right)_k$, and $\hat{w}_k$ were maximized using the EM algorithm [1][52] which was nested within the larger parameter training algorithm explained here. Since the EM for the mixture of geometric distributions was an iterative algorithm within another algorithm, the estimation of a after l^{th} iteration of the EM within the k^{th} iteration of the larger parameter training algorithm was denoted as $\hat{a}_{kl}$. Following this, the final solution of the k^{th} iteration of the EM on the mixture of geometric distributions was denoted as $\left(\hat{a}\right)_k$. In addition, $p(d_h, v_h = 1|\varphi_1, \varphi_2, w)$ and $p(d_h, v_h = 0|\varphi_1, \varphi_2, w)$ were used for brevity which were listed as (4.48) and (4.49) respectively. As with (4.30), the EM algorithm is iterative and convergence was measured using the same metric. Following this, to initiate the algorithm, $\left|\left|\{(\hat{\varphi}_1)_{k0}, (\hat{\varphi}_2)_{k0}, \hat{w}_{k0}\} - \left\{(\hat{\varphi}_1)_{k(-1)}, (\hat{\varphi}_2)_{k(-1)}, \hat{w}_{k(-1)}\right\}\right|\right|_\infty$ was set to a arbitrary value greater than δ. For a more detailed description on using the *EM* method to find the MLE of a mixture of two geometric distributions, see Section 2.1.

$$p(d_h, v_h = 1|\varphi_1, \varphi_2, w) = w(1-\varphi_1)^{d_h-1}\varphi_1 \tag{4.48}$$

$$p(d_h, v_h = 0|\varphi_1, \varphi_2, w) = (1-w)(1-\varphi_2)^{d_h-1}\varphi_2 \tag{4.49}$$

EM Algorithm for a mixture of two Geometric Distributions (4.50)

```
I Let $\{(\hat{\varphi}_1)_{k0}, (\hat{\varphi}_2)_{k0}, \hat{w}_{k0}\} = \left\{(\hat{\varphi}_1)_{(k-1)}, (\hat{\varphi}_2)_{(k-1)}, \hat{w}_{(k-1)}\right\}$
II Set $l = 0$ and define $\left|\left|\{(\hat{\varphi}_1)_{k0}, (\hat{\varphi}_2)_{k0}, \hat{w}_{k0}\} - \left\{(\hat{\varphi}_1)_{k(-1)}, (\hat{\varphi}_2)_{k(-1)}, \hat{w}_{k(-1)}\right\}\right|\right|_\infty > \delta$.
III While $\left|\left|\{(\hat{\varphi}_1)_{kl}, (\hat{\varphi}_2)_{kl}, \hat{w}_{kl}\} - \left\{(\hat{\varphi}_1)_{k(l-1)}, (\hat{\varphi}_2)_{k(l-1)}, \hat{w}_{k(l-1)}\right\}\right|\right|_\infty > \delta$
    (a) $l = l + 1$
    (b) for $h = 1 : n$
        i. $(\mathbb{E}[v_h])_{kl} = \frac{p\left(v_k(d_h), v_h=1\middle|(\hat{\varphi}_1)_{k(l-1)}, (\hat{\varphi}_1)_{k(l-1)}, \hat{w}_{k(l-1)}\right)}{\sum_{f=1}^{2} p\left(v_k(d_h), v_h=f\middle|(\hat{\varphi}_1)_{k(l-1)}, (\hat{\varphi}_1)_{k(l-1)}, \hat{w}_{k(l-1)}\right)}$
    (c) $(\hat{\varphi}_1)_{kl} = \frac{\sum_{h=1}^{n} (\mathbb{E}[v_h])_{kl}}{\sum_{h=1}^{n} (\mathbb{E}[v_h])_{kl}(v_k(d_h))}$
    (d) $(\hat{\varphi}_2)_{kl} = \frac{\sum_{h=1}^{n} 1-(\mathbb{E}[v_h])_{kl}}{\sum_{h=1}^{n} \left(1-(\mathbb{E}[v_h])_{kl}\right)(v_k(d_h))}$
    (e) $\hat{w}_{kl} = \frac{\sum_{h=1}^{n} (\mathbb{E}[v_h])_{kl}}{n}$
    (f) $\{(\hat{\varphi}_1)_k, (\hat{\varphi}_2)_k, \hat{w}_k\} = \{(\hat{\varphi}_1)_{kl}, (\hat{\varphi}_2)_{kl}, \hat{w}_{kl}\}$
```

Finally, the EM algorithm is also a local optimizer. In practice, using $(\hat{\varphi}_1)_{k-1}$, $(\hat{\varphi}_2)_{k-1}$, and $\hat{w}_{k-1}$ as the initial guess for the EM was sufficiently close to find the global optimum.

Performance

As with the two state system, two qualities of the algorithm were checked. The first was the convergence of the algorithm. This was done by exploring the behavior of the algorithm on the same dataset given multiple initial guesses. The second was to check the quality of inference. This was done by generating many datasets with known parameters and checking the estimates of the algorithm with the true generating parameters.

To check convergence, the algorithm was run three times on a dataset with $1,000,000$ observations where $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$. The initial guesses for μ_{S_1} and μ_{S_2} needed to be such that $\mu_{S_1} > \bar{\boldsymbol{y}}_1^T > \mu_{S_2}$. To achieve this $(\hat{\mu}_{S_1})_0$ was set equal to the 75 percentile of $\boldsymbol{y}_1^T$ and $(\hat{\mu}_{S_2})_0$ was set to 25 percentile of $\boldsymbol{y}_1^T$. For φ_1, φ_2 and w values above and below the true values were used for the initial guess. Since φ_1 and φ_2 can act in a similar fashion as μ_{S_1} and μ_{S_2}, a significant difference was maintained between $(\hat{\varphi}_1)_0$ and $(\hat{\varphi}_2)_0$. Finally, the initial guesses for q_{12} and σ^2 were generated randomly. The path of $\hat{\boldsymbol{\theta}}_j$, except μ_{S_1} and μ_{S_2} were placed in Figure 4.8. The paths of $(\mu_{S_1})_j$ and $(\mu_{S_2})_j$ were omitted as the starting point was the same each time.

In Figure 4.8, it was observed that all initial guesses converged to $\hat{q}_{12} = 0.000976$, $\hat{\varphi}_1 = 0.0101$, $\hat{\varphi}_2 = 0.0020$, $\hat{w} = 0.8739$, and $\sigma^2 = 8.9950$. In addition, $\hat{\mu}_{S_1}$ and $\hat{\mu}_{S_2}$ converged to 31.9999 and 25.9910 respectively. Like observed with Viterbi Training, the parameter training algorithm introduced in Section 4.2.2 re-

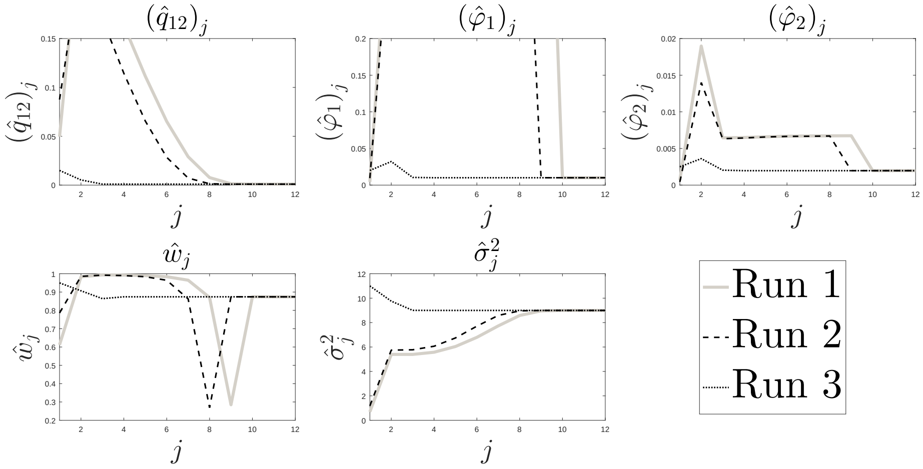


Figure 4.8: Comparing $\frac{\varphi_1}{\varphi_2}$ to r_{23} and r_{32} when $r_{21} = 1000$ and $\Delta t = \frac{1}{100000}$

quires better guesses than the Baum-Welch algorithm. However, given reasonable guesses as observed in Figure 4.8, the algorithm did converge.

Since the transition rates were the primary interest, the second test was to test the values of $\hat{r}_{ij}$ against the true values of r_{ij}. As stated previously, having observed similar behavior to Viterbi training, it was expected that the parameter training algorithm developed would introduce bias. To test this and evaluate whether the bias was problematic, the parameter training algorithm developed was run on 1136 computer generated datasets with with $1,000,000$ observations where $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$. The values of r_{12} were varied from 60 to 180, values of r_{21} were varied from 475 to 2025, values of r_{23} were varied from 40 to 276, and values of r_{32} were varied from 43 to 590. To avoid the problems discussed in Section 4.2.1, the majority of the values of r_{ij} were taken from the center of the region. Furthermore, the values of each r_{ij} were randomly assembled with values of r_{mn}. Since most values were picked in the middle of their respective ranges, it was very unlikely that values from the edge of the region in one dimension were assembled with a value near the edge of another dimension.

To assess the results, relative error or $\frac{\hat{r}_{ij}-r_{ij}}{r_{ij}}$ was used.

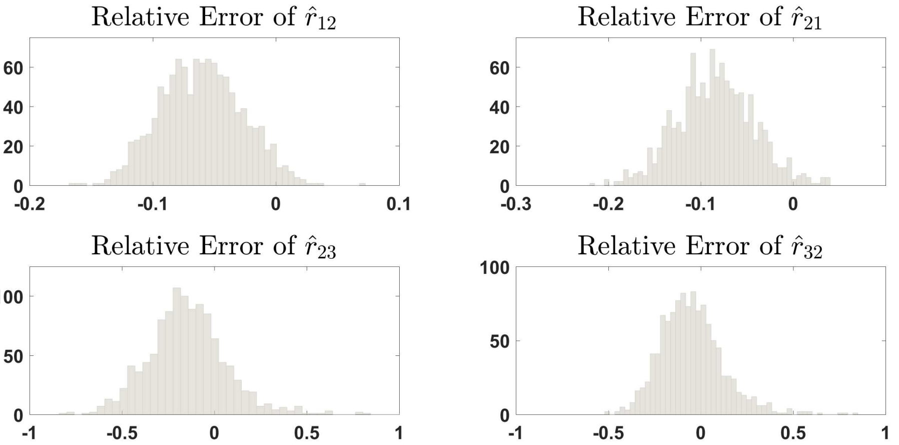


Figure 4.9: Histograms of relative error or $\frac{\hat{r}_{ij}-r_{ij}}{r_{ij}}$ for the parameter training algorithm developed on 1136 computer generated datasets with $1,000,000$ observations

Figure 4.9 has histograms of the relative error of $\hat{r}_{ij}$ estimated by the parameter training algorithm. The relative error of $\hat{r}_{12}$ varied from about -0.15 to 0.05. Since the mean of the relative error approximately -0.06, the bias was a sizable contributor to the error. The relative error of $\hat{r}_{21}$ was slightly larger, and the mean of the relative error approximately -0.085. Again, the bias had an obvious contribution to the error. The ranges of the relative error of $\hat{r}_{23}$ and $\hat{r}_{32}$ were much larger. This was expected as there was very little information on transition between states S_{2A} and S_{2B}. However, the mean of the relative error of $\hat{r}_{23}$ was approximately -0.15. This was still noticeable given the range. The contribution of the bias to the relative error of $\hat{r}_{32}$ was less noticeable. The mean of the relative error of $\hat{r}_{32}$ was approximately -0.05 which was less substantial given the range of the relative error of $\hat{r}_{32}$.

For most transitions rates, the bias was a noticeable contributor to the relative error. This was not surprising given the low amount of information provided by the data and the very low transition probabilities involved. Therefore, methods were explored to reduce the bias introduced by the algorithm.

4.2.3 Bias Reduction

In Section 4.2.2, it was seen that the parameter training algorithm developed, exhibited bias for inference on the transition rates. Furthermore, the bias made a noticeable contribution to the error and therefore effort to reduce the bias was prudent.

The results of the algorithm introduced in Section 4.2.2 on computer generated datasets discussed in Section 4.2.2 provided a possible solution to the issue presented by the bias of the algorithm. The results of the inference on the computer generated data created a set of pairs, $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$. In turn, $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$ was used to create an additional model to reduce this bias. This process fits under a methodology often referred to as computer experiments. Computer modeling and experiments has been an area of focus over the last few decades as compute power and resources have become more widely accessible [55]. This research differs slightly from the most common application of computer experiments. Computer experiments were often based on a computationally expensive simulation and running that code for all points of interest was considered too expensive. Instead, the data collected through the computer experiment was used to build a computationally less expensive approximating function (sometimes referred to as a meta-model) that can predict the relationships at points of interest. In this case, the simulations were only moderately expensive as making inference on a dataset with 1,000,000 datapoints took from 4-10 minutes. Furthermore, since each simulation was independent of each other, the operation was embarrassingly parallel. Therefore, compute time was not problematic. However, Kennedy and O'Hagan suggested that these methods can also be used to address “model inadequacy” or the difference between the true mean and the output of the algorithm [27].

Let the result of the algorithm be denoted as $\hat{\boldsymbol{\theta}} = PT\left(\boldsymbol{y}_1^T\right)$ where PT was the algorithm developed and $\boldsymbol{y}_1^T$ was the electric signal observed at $\boldsymbol{t} = (t_1, t_2, ..., t_T)$. Furthermore, $\boldsymbol{y}_1^T$ was a non-deterministic function of the true value of $\boldsymbol{\theta}$ which was denoted as $\boldsymbol{y}_1^T = g(\boldsymbol{\theta})$ where g was the Gillespie algorithm for a 3-state system presented in Section 4.1.1 used to generate $\boldsymbol{y}_1^T$. Therefore, $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ have the relationship listed in (4.51). Since $g(\boldsymbol{\theta})$ introduced randomness, (4.51) was re-written in (4.52) where $\mu(\boldsymbol{\theta})$ denotes the bias or non-random part of (4.51) and ε denotes the white noise.

$$\hat{\boldsymbol{\theta}} = PT\left(g\left(\boldsymbol{\theta}\right)\right) \tag{4.51}$$

$$\hat{\boldsymbol{\theta}} = \mu\left(\boldsymbol{\theta}\right) + \varepsilon \tag{4.52}$$

$$\varepsilon \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

The goal of this research was to make inference on $\boldsymbol{\theta}$ when the data was not generated by the computer. In this case, the true $\boldsymbol{\theta}$ was unknown. This pair of $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ was referred to as the predictive pair and was denoted as $\left(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_*\right)$. For this, given enough well placed pairs of $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$ from computer generated data sets, inference on $\mu(\boldsymbol{\theta})$ and σ_ε^2 was made. Since $\hat{\boldsymbol{\theta}}_*$ was known after running the training algorithm on data, predictive inference on $\boldsymbol{\theta}_*$ was made by considering the inverse problem of (4.52).

To make inference on $\boldsymbol{\theta}_*$, first a meta-model of the unknown function listed as (4.52) was built. The Gaussian process is a common choice to model an unknown function due to its flexibility [40][27] and following the paper ,“Design and Analysis of Computer Experiments” by Sacks and his colleagues [49], Gaussian processes have become a standard for the meta-model of a computer experiment [21]. Unfortunately, Gaussian processes are a computationally expensive choice for meta-model and should not be applied if a simpler meta-model has similar performance. However, the number of $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$ pairs needed for this application was relatively small and the Gaussian process meta-model had reasonable compute times. Furthermore, in Section 4.2.6, it can be seen that inference on the parameters of the meta-model indicated that there was some non-linear relation between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$, thus necessitating the use of Gaussian process model or some other non-linear model. Therefore, the meta model for (4.52), partially defined in (4.53)-(4.55) was applied where $\mathcal{GP}$ denotes a Gaussian Process and $s^2\boldsymbol{C}$ denotes the covariance function. The full definition of the Bayesian Gaussian process applied was defined in the next section. Rassmussen's lecture entitled Gaussian Processes in Machine Learning [7] and the similarly titled book [45] by himself and Christopher Williams are good introductions to Gaussian processes. [28], [55] and [51] are books on computer experiments that have more detailed explanations of Gaussian processes in that context.

$$\hat{\boldsymbol{\theta}} = \underbrace{\boldsymbol{\theta}\boldsymbol{\beta} + f(\boldsymbol{\theta})}_{\mu(\boldsymbol{\theta})} + \boldsymbol{\varepsilon} \tag{4.53}$$

$$\varepsilon \sim \mathcal{N}(0, \underbrace{s^2\nu^2}_{\sigma^2_\varepsilon}) \tag{4.54}$$

$$f(\boldsymbol{\theta}) \sim \mathcal{GP}(0, s^2 C(\boldsymbol{\theta}, \boldsymbol{\theta})) \tag{4.55}$$

4.2.4 A brief introduction to Gaussian process regression

Given a time series of the type described in Section 4.1, the estimator of the true parameters using the parameter training introduced in Section 4.2.2 was biased. By generating a list of $(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ pairs, the bias could be described by a meta-model which in turn could be used to reduce the aforementioned bias. A Bayesian Gaussian process was used as the meta-model where the posterior was simulated using an $MCMC$ sampler. The Gaussian process, originally referred to as kriging, was first introduced in a 1963 paper by Georges Matheron [35]. The original paper used kriging as a means for spatially modeling ore grades based on previously observed extractions. Due to the flexibility of kriging or Gaussian processes, the applications in geology and computer experiments are two of many possible uses. Since the Gaussian process described next has so many application beyond what was described here, the standard notation for predictor $(\boldsymbol{x}_i)$ and response (y_i) were used where there were N observations.

Although the problem introduced has a multivariate predictor and response, a univariate response was introduced first. For the model applied here, let y_i be a univariate response and $\boldsymbol{x}_i$ be a vector of k predictor variables. The full definition was listed as (4.56) to (4.62) where MVN denotes the multivariate normal distribution and $C(\boldsymbol{x}, \boldsymbol{x})$ is the squared exponential kernel.

$$\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\varepsilon} \tag{4.56}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_N \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ . \\ . \\ \boldsymbol{x}_N \end{bmatrix} \qquad \boldsymbol{x}_i = \begin{bmatrix} x_{i1} & x_{i2} & . & . & . & x_{ik} \end{bmatrix} \tag{4.57}$$

$$\boldsymbol{f(X)} = \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ . \\ . \\ f(\boldsymbol{x}_N) \end{bmatrix} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ \varepsilon_N \end{bmatrix} \qquad \varepsilon_i \sim N(0, \sigma_\varepsilon^2 = s^2\nu^2) \tag{4.58}$$

$$\boldsymbol{f(X)}|\boldsymbol{X}, \vartheta \sim MVN\Big(\boldsymbol{0}, s^2 C(\boldsymbol{x}, \boldsymbol{x})\Big) \tag{4.59}$$

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\sum_{h=1}^{k} \frac{(x_{ih} - x_{jh})^2}{l_h^2}\right\} \tag{4.60}$$

$$\boldsymbol{y}|\boldsymbol{X}, \beta, \boldsymbol{l}^2, s^2, \nu^2 \sim MVN\Big(\boldsymbol{X\beta}, s^2 C(\boldsymbol{x}, \boldsymbol{x}) + s^2\nu^2 I\Big) \tag{4.61}$$

$$C(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} c(\boldsymbol{x}_1, \boldsymbol{x}_1) & c(\boldsymbol{x}_1, \boldsymbol{x}_2) & . & . & . & c(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ c(\boldsymbol{x}_2, \boldsymbol{x}_1) & . & . & . & . & c(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ c(\boldsymbol{x}_N, \boldsymbol{x}_1) & . & . & . & . & c(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \tag{4.62}$$

(4.56) to (4.62) has two uncorrelated parts. The first was the linear regression or $\boldsymbol{y} - \boldsymbol{X\beta}$. For the problem introduced in this research, this has a practical interpretation. Since $\hat{\boldsymbol{\theta}}$ was a bias estimator of $\boldsymbol{\theta}$, it was expected that there should be a correlation between the true value of $\boldsymbol{\theta}$ and the corresponding $\hat{\boldsymbol{\theta}}$. β provides a very straight forward interpretation of that. The second part $(\boldsymbol{f(x)} + \boldsymbol{\varepsilon})$ was a model of the residuals of the linear regression. This particular version is sometimes referred to as the nugget model. Often when using the nugget model, it is defined as one term $\boldsymbol{f(X)}$ where $\boldsymbol{f(X)}$ is a zero mean Gaussian process with variance

$s^2 C(\boldsymbol{x}, \boldsymbol{x}) + s^2 \nu^2 \boldsymbol{I}$. However, for the purpose of this research $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma_\varepsilon^2 = s^2\nu^2)$ was expressed separately since the aforementioned experiment had randomness and therefore the noise term had an explicit meaning.

The use of the nugget model for this application was necessary as $\hat{\boldsymbol{\theta}}$ was a stochastic realization of the explanatory variable, $\boldsymbol{\theta}$. However, in the general case, there are many reasons to apply the nugget model even if the relation between the explanatory and response variables were deterministic. For a discussion on the application of the nugget model in a deterministic setting, see Gramacy's paper "Cases for the nugget in modeling computer experiments" [21].

The goal was to generate a less biased estimator of $\boldsymbol{\theta}$ than $\hat{\boldsymbol{\theta}}$ produced from the algorithm introduced in Section 4.2.2. For this a computer generated dataset of $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ pairs which are synonymous with $\boldsymbol{x}_i$ and y_i respectively. This dataset was used to tune the parameters $\boldsymbol{l}^2$, s^2, ν^2, and $\boldsymbol{\beta}$ which were referred to as ϑ to avoid confusion with the parameters of the hidden Markov model. Given the inference on ϑ, inference could be made on an unknown $\boldsymbol{\theta}_*$ or $\boldsymbol{x}_*$ given $\hat{\boldsymbol{\theta}}_*$ or y_* calculated by the parameter training algorithm from Section 4.2.2. This was done using a Gibbs sampler to create posteriors of $\boldsymbol{l}^2$, s^2, ν^2, $\boldsymbol{\beta}$, and $\boldsymbol{x}_*$. As before, the i^{th} draw from the posterior of k was denoted by k_i or $(k_h)_i$ if k had a subscript and ϑ_{-k} denoted all elements of ϑ except k. A sampling algorithm for D draws from the posterior when given one response variable was listed as (4.63). It was important to note that for some of the conditional posteriors, two conditional priors were presented. For those conditional posteriors, the sampler presents two possible formulations to choose from.

A sampler for $\boldsymbol{x}_* | y_*, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{l}^2, s^2, \beta$ (4.63)

```
I. Pick $\boldsymbol{l}_0^2$, $\beta_0$, and $(\boldsymbol{x}_*)_0$

II. For $i = 1 : D$
```

```
    (i) Draw $s_i^2$ from (4.64) or (4.65).
    (ii) Draw $\boldsymbol{\beta}_i$ from (4.68)
    (iii) For $m = 1 : k$
        A. Draw $(l_m^2)_p$ from (4.69) or (4.71).
        B. Set $r = \frac{p\left(\log\left(\left(l_m^2\right)_p\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}\right)/J\left(\log\left(\left(l_m^2\right)_p\right)|\log\left(\left(l_m^2\right)_{i-1}\right)\right)}{p\left(\log\left((l_m^2)_{i-1}\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}\right)/J\left(\log\left((l_m^2)_{i-1}\right)|\log\left((l_m^2)_p\right)\right)}$ using (4.69) and (4.70) or (4.71) and (4.72).
        C. Get $c$ where $c \sim unif(0,1)$
        D. If $r \geq c$ than $\log\left((l_m^2)_i\right) = \log\left(\left(l_m^2\right)_p\right)$ otherwise $\log\left((l_m^2)_i\right) = \log\left(\left(l_m^2\right)_{i-1}\right)$
    (iv) end
    (v) Draw $\nu_p^2$ from (4.73) or (4.75).
    (vi) Set $r = \frac{p\left(\log\left(\nu_p^2\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-\nu^2}\right)/J\left(\log\left(\nu_p^2\right)|\log\left(\nu_{i-1}^2\right)\right)}{p\left(\log\left(\nu_{i-1}^2\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-\nu^2}\right)/J\left(\log\left(\nu_{i-1}^2\right)|\log(\nu_p^2)\right)}$ using (4.73) and (4.74) or (4.75) and (4.76).
    (vii) Get $c$ where $c \sim unif(0,1)$
    (viii) If $r \geq c$ than $\log(\nu_i^2) = \log(\nu_p^2)$ otherwise $\log(\nu_i^2) = \log(\nu_{i-1}^2)$
    (ix) For $m = 1 : k$
        A. Draw $(x_{*m})_p$ from (4.77).
        B. Set $r = \frac{p\left((x_{*m})_p|\boldsymbol{y},\boldsymbol{X},\boldsymbol{y}_*,(\boldsymbol{x}_*)_{-x_{*m}},\vartheta\right)/J\left((x_{*m})_p|(x_{*m})_{i-1}\right)}{p\left((x_{*m})_{i-1}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{y}_*,(\boldsymbol{x}_*)_{-x_{*m}},\vartheta\right)/J\left((x_{*m})_{i-1}|(x_{*m})_p\right)}$ using (4.77) and (4.78).
        C. Get $c$ where $c \sim unif(0,1)$
        D. If $r \geq c$ than $(x_{m*})_i = (x_{m*})_p$ otherwise $(x_{m*})_i = (x_{m*})_{i-1}$
    (x) end
III. end
```

Draws from $s^2|\boldsymbol{y}, \boldsymbol{X}, \beta, \boldsymbol{l}^2, \nu^2$

Given the construction of the nugget model, a Gibbs sample was possible for the conditional posterior of s^2 with the correct choice of prior. Following this, a conditionally conjugate prior was used. In this case, the inverse gamma distribution was the conditionally conjugate prior for s^2 which was denoted as $\mathcal{IG}\left(\frac{n_{s2}}{2}, \frac{d_{s2}}{2}\right)$. If a less informative prior is desired, the improper case where $n_{s^2} = d_{s^2} = 0$ which is equivalent to $s^2 \propto \frac{1}{s^2}$ still yields an inverse gamma distribution for the posterior. However, Lawrence and his colleagues noted that efficient sampling, "has proved to be particularly difficult in many GP applications, because the posterior distribution describes a highly correlated high-dimensional variable" [29]. Given that, it may be desirable to use a more informative prior by choosing different n_{s^2}, d_{s^2}, or truncating the distribution if justifiable for the model. The truncated inverse gamma distribution with parameters $\frac{n}{2}$, $\frac{d}{2}$, and a maximum value of U was denoted as $\mathcal{TIG}\left(\frac{n}{2}, \frac{d}{2}, U\right)$. The resulting posteriors were listed as (4.64) to (4.66) where N was the size of the dataset, $\boldsymbol{C} = C(\boldsymbol{x}, \boldsymbol{x})$, and ϑ_{-k} denotes all parameters except k. Derivations of (4.64) to (4.66) can be found in Appendix D.1.1.

$$\begin{aligned}
\text{If } s^2 &\sim \mathcal{IG}\left(\frac{n_{s^2}}{2}, \frac{d_{s^2}}{2}\right) \\
s^2|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-s^2} &\sim \mathcal{IG}\left(\frac{N + n_{s^2}}{2}, \frac{\boldsymbol{S}^2 + d_{s^2}}{2}\right) \qquad (4.64) \\
\text{If } s^2 &\sim \mathcal{TIG}\left(\frac{n_{s^2}}{2}, \frac{d_{s^2}}{2}, U\right) \\
s^2|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-s^2} &\sim \mathcal{TIG}\left(\frac{N + n_{s^2}}{2}, \frac{\boldsymbol{S}^2 + d_{s^2}}{2}, U\right) \qquad (4.65) \\
\boldsymbol{S}^2 &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T \left(\boldsymbol{C} + \nu^2 I\right)^{-1} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \qquad (4.66)
\end{aligned}$$

Finally, many software packages do not have a truncated inverse gamma random number generator. Since, the truncated inverse gamma is proportional to an inverse gamma distribution over the support of truncated inverse gamma, draws from $\mathcal{TIG}\left(\frac{n}{2}, \frac{d}{2}, U\right)$ were made from $\mathcal{IG}\left(\frac{n_{s^2}}{2}, \frac{d_{s^2}}{2}\right)$. Draws from the conditional posterior greater than U were be rejected and another draw was made. If values greater than U were "rare" in the posterior, the speed should not be noticeably different from using the inverse gamma as a conditional prior. If draws greater than U occur frequently enough that the sampler was slowed, the use of the $\mathcal{TIG}\left(\frac{n}{2}, \frac{d}{2}, U\right)$ should be reviewed.

Draws from $\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, s^2, \boldsymbol{l}^2, \nu^2$

From (4.62), the likelihood of $\boldsymbol{y}$ was a multivariate normal distribution with mean $\boldsymbol{X\beta}$ and variance $s^2C(\boldsymbol{x}, \boldsymbol{x}) + s^2\nu^2\boldsymbol{I}$. Since $s^2C(\boldsymbol{x}, \boldsymbol{x}) + s^2\nu^2\boldsymbol{I}$ was not correlated with β, the conditional posterior of β was treated as a linear regression problem with known variance. In that case, the improper conditional prior $p(\beta) \propto 1$ resulted in a conditional posterior that could be drawn from a Gibbs sample. This prior was advantageous as it was non-informative and sampled efficiently. The conditional prior and the resulting conditional posterior was listed as (4.67) and (4.68) respectively. For brevity, let $\boldsymbol{\Sigma}_{GP} = s^2C(\boldsymbol{x}, \boldsymbol{x}) + s^2\nu^2\boldsymbol{I}$.

$$\boldsymbol{\beta} \propto 1 \tag{4.67}$$

$$\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta} \propto MVN\left(\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}, \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\right) \tag{4.68}$$

To sample effectively from (4.68) the standard numerical inversion methods could not be used. Since $\boldsymbol{\Sigma}_{GP}$ and $\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)$ were symmetric and positive definite, the matrices were inverted using Cholesky decomposition. Furthermore, the symmetric and positive definite properties were not always maintained numerically.

Therefore, a function called nearestSPD (nearest symmetric positive definite matrix) based on the work of Higham was applied [12] [23]. Alternatively, Gelman and his colleagues suggest a slightly different method to address these issues which can be found in [17]. A derivation of (4.68) can be found in append D.1.2.

Draws from $l_m^2|\boldsymbol{y}, \boldsymbol{X}, s^2, \boldsymbol{\beta}, \nu^2, \boldsymbol{l}^2_{-l_m^2}$

Unlike s^2 and $\boldsymbol{\beta}$, there was not a conditional prior such that a Gibbs sample can be taken of the conditional posterior. Therefore, a Metropolis-Hasting sample as described in Section 2.2 was used. Following this, the inverse gamma distribution was used as the conditional prior as it had the same supports as l_m^2 and the amount of information that was added by the conditional prior could be varied through the parameters. Furthermore, since l_m^2 represents the length scale, in many cases it should be limited to a number not much bigger than the size of the region. Therefore, in some applications, the truncated inverse gamma distribution was a better choice, and the ease of application was already discussed in Section 4.2.4. Another consideration was the dimension of $\boldsymbol{x}$. If the h^{th} dimension had little or no effect on $\boldsymbol{y}$, the nugget model does not have a variable that is interpreted as the scale factor for each dimension. However, Neal noted that the length scale can control the degree of relevance of each dimension. If the length scale for the h^{th} dimension was very large, the input in the h^{th} dimension had little influence on the amount of correlation between two inputs [39].

To fit all possible situations, a Metropolis-Hasting "jumping distribution" denoted as $J\left(\log(l_p^2)|\log((l_m^2)_{i-1}\right)$ was listed as (4.69) and (4.71) for both an inverse gamma and a truncated inverse gamma respectively. The corresponding conditional posteriors were listed as (4.70) and (4.72) respectively. To restrict draws to the support of l_m^2, draws were made from $\log(l_m^2)$. In the case the conditional prior was an inverse gamma distribution, the “jumping distribution” was a normal distribution. For the case that the conditional prior was the truncated inverse gamma distribution, a truncated normal distribution was used where $\mathcal{TN}(\mu, \sigma^2, [a, b])$ denotes a truncated normal with mean μ, variance σ^2, and was restricted to the interval $[a, b]$. As before, let $\boldsymbol{\Sigma}_{GP} = s^2 C(\boldsymbol{x}, \boldsymbol{x}) + s^2\nu^2\boldsymbol{I}$ and derivations of the following can be found in Appendix D.1.3.

$$\text{The } i^{th}\text{draw if } l_m^2 \sim \mathcal{IG}\left(n_{l_m^2}, d_{l_m^2}\right)$$

$$\log\left(\left(l_m^2\right)_p\right) \sim \mathcal{N}\left(\left(\log\left(l_m^2\right)_{i-1}\right), \tau_{l^2}^2\right) \tag{4.69}$$

$$p(\log\left(l_m^2\right)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-l_m^2}) \propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right)^T \left(\boldsymbol{\Sigma}_{GP}\right)^{-1} \cdots\right. \tag{4.70}$$

$$\left.\cdots \times \left(\boldsymbol{y} - \boldsymbol{X\beta}\right)\right\} \left(l_m^2\right)^{-n_{l_m^2}-1} \exp\left\{-\frac{d_{l_m^2}}{l_m^2}\right\} \times l_m^2$$

$$\text{The } i^{th}\text{draw if } l_m^2 \sim \mathcal{TIG}\left(n_{l_m^2}, d_{l_m^2,}, U\right)$$

$$\log\left(\left(l_m^2\right)\right) \sim \mathcal{TN}\left(\log\left(\left(l_m^2\right)_{i-1}\right), \tau_{l^2}^2, (-\infty, \log(U)]\right) \tag{4.71}$$

$$p(\log\left(l_m^2\right)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-l_m^2}) \propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right)^T \left(\boldsymbol{\Sigma}_{GP}\right)^{-1} \cdots\right. \tag{4.72}$$

$$\left.\cdots \times \left(\boldsymbol{y} - \boldsymbol{X\beta}\right)\right\} \left(l_m^2\right)^{-n_{l_m^2}} \exp\left\{-\frac{d_{l_m^2}}{l_m^2}\right\} \times I\left[-\infty < \log\left(l_m^2\right) \leq \log(U)\right]$$

Draws from $\nu^2|\boldsymbol{y}, \boldsymbol{X}, s^2, \boldsymbol{\beta}, \boldsymbol{l}^2$

As with l_m^2, a Metropolis-Hasting sample from the conditional posterior was necessary. Again, the inverse gamma or truncated inverse gamma was used as the conditional prior because of the versatility discussed in Section 4.2.4. Although ν^2 did not have a straight forward interpretation, $s^2\nu^2 = \sigma_\varepsilon^2$ or $s^2\nu^2$ was the noise of the problem. By limiting ν^2, it keeps σ_ε^2 from becoming too large or s^2 from becoming too small which may contradict prior information. The jumping distributions were listed as (4.73) and (4.75) for the inverse gamma and truncated inverse gamma conditional priors respectively. The corresponding posteriors were listed as (4.74) and (4.76) and the derivation can be found in Appendix D.1.4.

$$\begin{aligned}
&\text{The } i^{th}\text{draw if } \nu^2 \sim \mathcal{IG}\left(n_{\nu^2}, d_{\nu^2}\right) \\
&\log\left(\nu_p^2\right) \sim \mathcal{N}\left(\log\left(\nu_{i-1}^2\right), \tau_{\nu^2}^2\right) && (4.73) \\
&p(\log\left(\nu^2\right)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}) \propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^T \left(\boldsymbol{\Sigma}_{GP}\right)^{-1} \cdots\right. && (4.74) \\
&\left.\cdots \times \left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)\right\} \left(\nu^2\right)^{-n_{\nu^2}-1} \exp\left\{-\frac{d_{\nu^2}}{\nu^2}\right\} \times \nu^2 \\
&\text{The } i^{th}\text{draw if } \nu^2 \sim \mathcal{TIG}\left(n_{\nu^2}, d_{\nu^2}, U\right) \\
&\log\left(\nu_p^2\right) \sim \mathcal{TN}\left(\log\left(\nu_{i-1}^2\right), \tau_{\nu^2}^2, (-\infty, \log(U)]\right) && (4.75) \\
&p(\log\left(\nu^2\right)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}) \propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^T \left(\boldsymbol{\Sigma}_{GP}\right)^{-1} \cdots\right. && (4.76) \\
&\left.\cdots \times \left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)\right\} \left(\nu^2\right)^{-n_{\nu^2}} \exp\left\{-\frac{d_{\nu^2}}{\nu^2}\right\} \times I\left[-\infty < \log\left(\nu^2\right) \le \log(U)\right]
\end{aligned}$$

Draws from $x_{*m}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{y}_*, \left(\boldsymbol{x}_*\right)_{-x_{*m}}, \vartheta$

Drawing from x_{*m} was also a Metropolis-Hasting sample from the conditional posterior. A conditional prior was suggested here and a second will be suggested in terms of meta-model for $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$. The first conditional prior was built on the assumption that the dataset was collected around an estimated point for $\boldsymbol{x}_*$. Therefore, it was reasonable to assume that the true x_{*m} was near the average of the m^{th} dimension of the collected observations or $\frac{\sum_{i=1}^N x_{im}}{N}$. Furthermore, since it was necessary that x_{m*} was in the range of all observations collected to make inference, it was assumed that $x_{m*} \in [a, b]$ where $[a, b]$ was the range of all collected data in the m^{th} dimension. From this, a truncated normal distribution restricted to $[a, b]$ with mean $\frac{\sum_{i=1}^N x_{im}}{N}$ and variance τ_*^2 was used for the conditional prior where a large τ_*^2 signifies less confidence in the prior assertion. The jumping distribution for the Metropolis-Hasting sample was a truncated normal distribution listed as (4.77), and the corresponding posterior was listed as (4.78).

$$\text{The } i^{th}\text{draw if } x_{*m} \sim \mathcal{TN}\left(\frac{\sum_{i=1}^{N} x_{im}}{N}, \tau_*^2, [a,b]\right)$$

$$\left(x_{*m}\right)_p \sim \mathcal{N}\left(\left(x_{*m}\right)_{i-1}, \tau^2_{x_{*m}}\right) \tag{4.77}$$

$$p\left(x_{*m} \,|\, \boldsymbol{y}, \boldsymbol{X}, y_*, (\boldsymbol{x}_*)_{-x_{*m}}, \vartheta\right) \propto \frac{1}{\sigma_{y_*}} \exp\left\{-\frac{\left(y_* - m_{y_*}\right)^2}{2\sigma^2_{y_*}}\right\}\cdots \tag{4.78}$$

$$\cdots \times \frac{1}{\tau_*} \exp\left\{-\frac{\left(x_{*m} - \frac{1}{N}\left(\sum_{i=1}^{N} x_{im}\right)\right)^2}{2\tau_*^2}\right\} \times I[a \leq q_{ij} \leq b]$$

$$m_{y_*} = s^2 C_*^T (s^2 C + s^2 \nu^2 I)^{-1} (\boldsymbol{y} - \boldsymbol{X\beta}) \tag{4.79}$$

$$\sigma^2_{y_*} = s^2 C_{**} - s^2 C_*^T \left(s^2 \nu^2 I + s^2 C\right)^{-1} s^2 C_* + s^2 \nu^2 I \tag{4.80}$$

$$C_* = \begin{bmatrix} c(\boldsymbol{x}_1, \boldsymbol{x}_*) & c(\boldsymbol{x}_2, \boldsymbol{x}_*) & . & . & . & c(\boldsymbol{x}_N, \boldsymbol{x}_*) \end{bmatrix}^T \tag{4.81}$$

A derivation for (4.79) and (4.80) can be found in Appendix D.1.5. (4.78) can be found in Appendix D.1.5.4. For inference on one $\boldsymbol{x}_*$ as discussed here, C_* was a $N \times 1$ vector and C_{**} was a scalar. However, it is possible to consider multiple predictive points at once. The formulas for (4.79) and (4.80) are the same and the first distribution in (4.78) would be a multivariate normal with mean m_{y_*} and covariance $\boldsymbol{\sigma}^2_{\boldsymbol{y}_*}$. For generality, the derivations in Appendices D.1.5 and D.1.5.4 were done using multiple predictive points. If it was desired to make inference on W different observations of $\boldsymbol{x}$, then C_* was a $N \times W$ matrix. Furthermore, C_{**} was a $W \times W$ matrix defined similar to (4.61) except it computes the correlation between all the different observations of $\boldsymbol{x}_*$.

4.2.5 Applying Gaussian processes to the parameter training algorithm introduced in section 4.2.2

It was shown that $\hat{\boldsymbol{\theta}}_*$ calculated by the algorithm introduced in Section 4.2.2 was a biased estimate for $\boldsymbol{\theta}_*$. To correct this bias, a set of computed simulated pairs of $(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ were generated. Using the MCMC sampler below, a meta-model was built, and the inverse problem was solved to find a less bias estimate of $\boldsymbol{\theta}_*$. However, the relative errors of the estimates of σ^2, $\mu_{S_1}^2$ and $\mu_{S_2}^2$ were much smaller than the estimates of the transition probabilities and consequently the transition rates. In addition, σ^2, $\mu_{S_1}^2$ and $\mu_{S_2}^2$ were part of the experimental setup but σ^2, $\mu_{S_1}^2$ and $\mu_{S_2}^2$ do not exist in vivo. Therefore, reducing the bias of inference made on σ^2, $\mu_{S_1}^2$ and $\mu_{S_2}^2$ was of less interest. Following this, bias reduction was run only on $\boldsymbol{\theta}_q = (q_{12}, q_{21}, q_{23}, q_{32})$ to reduce computation time.

The Gaussian process introduced in Section 4.2.4 had multiple inputs but a single response variable. However, $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{q_{ij}}\right)$ had four predictor and response varaibles. For this, it was assumed that the bias for each $\hat{q}_{ij}$ was independent. This was an equivalent statement to $\hat{q}_{ij}$ given $\boldsymbol{\theta}_q$ was conditionally independent of all $\hat{q}_{kl}$ where $ij \neq kl$. Therefore, inference on ϑ for each $\hat{q}_{ij}$ could be done separately following the same steps as 4.63 where $\hat{q}_{ij}$ corresponds with $\boldsymbol{y}$ and $\boldsymbol{\theta}_q$ corresponds with $\boldsymbol{X}$. To avoid confusion, the parameters associated with $\hat{q}_{ij}$ were denoted as $\vartheta_{q_{ij}} = \left(\boldsymbol{\beta}_{q_{ij}}, \boldsymbol{l}_{q_{ij}}^2, s_{q_{ij}}^2, \nu_{q_{ij}}^2\right)$ and $\vartheta = (\vartheta_{q_{12}}, \vartheta_{q_{21}}, \vartheta_{q_{23}}, \vartheta_{q_{32}})$. In the case of multidimensional parameters like $\boldsymbol{l}_{q_{ij}}^2$, the m^{th} dimension was denoted as $l_{q_{ij},m}^2$. The revised sampling algorithm for a multivariate response was listed as 4.82.

$$\text{A sampler for } \boldsymbol{\theta}_{*q} | \hat{\boldsymbol{\theta}}_{*q}, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_q, \vartheta \quad (4.82)$$

A. Pick ϑ_0 and $\boldsymbol{\theta}_{*q}$

B. For $i = 1 : D$

```
I. For all $q_{ij}$ in $\boldsymbol{\theta}_q$
    (i) Draw $\left(s^2_{q_{ij}}\right)_i$ from (4.64) or (4.65).
    (ii) Draw $\left(\boldsymbol{\beta}_{q_{ij}}\right)_i$ from (4.68)
    (iii) For $m = 1 : k$
        a. Draw $\left(l^2_{q_{ij},m}\right)_p$ from (4.69) or (4.71).
        b. Set $r = \dfrac{\dfrac{p\left(\log\left(\left(l^2_{q_{ij},m}\right)_p\right)|\hat{\boldsymbol{q}}_{ij},\boldsymbol{\theta}_{q_{ij}},\vartheta_{q_{ij},-l^2_{q_{ij},m}}\right)}{J\left(\log\left(\left(l^2_{q_{ij},m}\right)_p\right)|\log\left(\left(l^2_{q_{ij},m}\right)_{i-1}\right)\right)}}{\dfrac{p\left(\log\left(\left(l^2_{q_{ij},m}\right)_{i-1}\right)|\hat{\boldsymbol{q}}_{ij},\boldsymbol{\theta}_{q_{ij}},\vartheta_{q_{ij},-l^2_{q_{ij},m}}\right)}{J\left(\log\left(\left(l^2_{q_{ij},m}\right)_{i-1}\right)|\log\left(\left(l^2_{q_{ij},m}\right)_p\right)\right)}}$ using (4.69) and (4.70)
           or (4.71) and (4.72).
        c. Get $c$ where $c \sim unif(0,1)$
        d. If $r \geq c$ than $\log\left(\left(l^2_{q_{ij},m}\right)_i\right) = \log\left(\left(l^2_{q_{ij},m}\right)_p\right)$ otherwise $\log\left(\left(l^2_{q_{ij},m}\right)_i\right) =$
           $\log\left(\left(l^2_{q_{ij},m}\right)_{i-1}\right)$
    (iv) end
    (v) Draw $\left(\nu^2_{q_{ij}}\right)_p$ from (4.73) or (4.75).
    (vi) Set $r = \dfrac{p\left(\log\left(\left(\nu^2_{q_{ij}}\right)_p\right)|\hat{\boldsymbol{q}}_{ij},\boldsymbol{\theta}_{q_{ij}},\vartheta_{q_{ij},-\nu^2_{q_{ij}}}\right)/J\left(\log\left(\left(\nu^2_{q_{ij}}\right)_p\right)|\log\left(\left(\nu^2_{q_{ij}}\right)_{i-1}\right)\right)}{p\left(\log\left(\left(\nu^2_{q_{ij}}\right)_{i-1}\right)|\hat{\boldsymbol{q}}_{ij},\boldsymbol{\theta}_{q_{ij}},\vartheta_{q_{ij},-\nu^2_{q_{ij}}}\right)/J\left(\log\left(\left(\nu^2_{q_{ij}}\right)_{i-1}\right)|\log\left(\left(\nu^2_{q_{ij}}\right)_p\right)\right)}$
         using (4.73) and (4.74) or (4.75) and (4.76).
    (vii) Get $c$ where $c \sim unif(0,1)$
    (viii) If $r \geq c$ than $\log\left(\left(\nu^2_{q_{ij}}\right)_i\right) = \log\left(\left(\nu^2_{q_{ij}}\right)_p\right)$ otherwise $\log\left(\left(\nu^2_{q_{ij}}\right)_i\right) =$
           $\log\left(\left(\nu^2_{q_{ij}}\right)_{i-1}\right)$
    (ix) end
II. end
III. For all $q_{*ij}$ in $\boldsymbol{\theta}_{*q}$
    i. Draw $\left(q_{*ij}\right)_p$ from (4.83).
```

```
        ii. Set $r = \frac{p\left(\left(q_{*ij}\right)_p|\hat{\boldsymbol{\theta}}_q,\boldsymbol{\theta}_q,\hat{\boldsymbol{\theta}}_{*q},\boldsymbol{\theta}_{*q,-q_{*ij}},\vartheta\right)/J\left(\left(q_{*ij}\right)_p|\left(q_{*ij}\right)_{i-1}\right)}{p\left(\left(q_{*ij}\right)_{i-1}|\hat{\boldsymbol{\theta}}_q,\boldsymbol{\theta}_q,\hat{\boldsymbol{\theta}}_{*q},\boldsymbol{\theta}_{*q,-q_{*ij}},\vartheta\right)/J\left(\left(q_{*ij}\right)_{i-1}|\left(q_{*ij}\right)_p\right)}$ using (4.83) and (4.84).
        iii. Get $c$ where $c \sim unif(0,1)$
        iv. If $r \geq c$ than $\left(q_{*ij}\right)_i = \left(q_{*ij}\right)_p$ otherwise $\left(q_{*ij}\right)_i = \left(q_{*ij}\right)_{i-1}$
    IV. end
C. end
```

Drawing from q_{*ij} was different. Although it was assumed that each $\hat{q}_{*ij}$ given $\boldsymbol{\theta}_q$ was conditionally independent of the all other $\hat{q}_{*ij}$, the same was not true for all q_{ij}. Even with the assumed conditional independence, all q_{*ij} depended on all $\hat{q}_{*ij}$. This caused sampling from q_{*ij} to be slightly different than the sampler with a one dimensional response variable and was discussed further in Section 4.2.5.

Draws from $q_{*ij}|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta$

Like the case with the one dimensional response variable, drawing from the conditional posterior of q_{*ij} was a Metropolis-Hastings step. Since $\hat{q}_{*ij}$ was an estimator of q_{*ij}, it was assumed that $\hat{q}_{*ij}$ was near the true q_{*ij}. Furthermore, since the dataset was created specifically to estimate q_{*ij}, it was assumed that q_{*ij} was within the range of the datapoints generated for the computer experiment. Following this, the conditional prior was a truncated normal distribution limited to $[a, b]$ with mean $\hat{q}_{*ij}$ and variance τ_*^2.

$$\text{The } i^{th} \text{ draw if } q_{*ij} \sim \mathcal{TN}\left(\hat{q}_{ij}, \tau_*^2, [a,b]\right)$$

$$\left(q_{*ij}\right)_p \sim \mathcal{N}\left(\left(q_{*ij}\right)_{i-1}, \tau^2_{q_{*ij}}\right) \tag{4.83}$$

$$p\left(\left(q_{*ij}\right)_p | \hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right) \propto I[a \leq q_{*ij} \leq b] \times \cdots \tag{4.84}$$

$$\cdots \times \left[\prod_{\forall \hat{q}_{*ij} \in \hat{\boldsymbol{\theta}}_{*q}} \frac{1}{\sigma_{q_{*ij}}} \exp\left\{-\frac{\left(\hat{q}_{*ij} - m_{\hat{q}_{*ij}}\right)^2}{2\sigma^2_{\hat{q}_{*ij}}}\right\}\right] \cdots$$

$$\cdots \times \frac{1}{\tau_*} \exp\left\{-\frac{\left(q_{*ij} - \hat{q}_{*ij}\right)^2}{2\tau_*^2}\right\}$$

$$m_{\hat{q}_{*ij}} = s^2_{q_{ij}} C^T_{*q_{ij}} (s^2_{q_{ij}} C_{q_{ij}} + s^2_{q_{ij}} \nu^2_{q_{ij}} I)^{-1} \left(\hat{\boldsymbol{q}}_{ij} - \boldsymbol{\theta_q} \boldsymbol{\beta_{q_{ij}}}\right) \tag{4.85}$$

$$\sigma^2_{\hat{q}_{*ij}} = s^2_{q_{ij}} C_{**q_{ij}} - s^2_{q_{ij}} C^T_{*q_{ij}} \left(s^2_{q_{ij}} \nu^2_{q_{ij}} I + s^2_{q_{ij}} C_{q_{ij}}\right)^{-1} s^2_{q_{ij}} C_{*q_{ij}} + s^2_{q_{ij}} \nu^2_{q_{ij}} I \tag{4.86}$$

As with the general case, this model can be easily changed to deal with multiple predictions. This was discussed further in Section 4.2.4. In addition, a derivation can be found in Appendix D.1.5.4.

4.2.6 Initial analysis of bias reduction

Given a dataset $\boldsymbol{y}_1^T$, the parameter training algorithm results in biased estimators of transition probabilities $\left(\hat{\boldsymbol{\theta}}_{*q}\right)$. To estimate the bias, a set of computer simulated pairs of $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$ were generated. A meta-model was generated for the computer simulated pair,s and the inverse problem was solved for $\boldsymbol{\theta}_{q*}$ using the methods described in Section 4.2.5. To test if the bias was reduced, the methods were tested on a simulated 10 second dataset where $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma = 3$, and was sampled at 100 kHz. This translates to $\boldsymbol{\theta}_{*q} \approx (0.0010, 0.0099, 0.0010, 0.0020)$ and the estimated values using the parameter training algorithm from Section 4.2.2 were $\hat{\boldsymbol{\theta}}_{*q} \approx (0.0010, 0.0090, 0.0008, 0.0022)$. To reduce the bias a set of 163 $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs were generated to train the parameters of (4.53) and make prediction on $\boldsymbol{\theta}_{*q}$. For the 163 pairs, the range of q_{12} was $[0.007, 0.0016]$, the range of q_{21} was $[0.0055, 0.0135]$, the range of q_{23} was $[0.0004, 0.0023]$, and the range of q_{32} was

$[0.0007, 0.0047]$. $30,000$ draws from the posterior was made using (4.82). The histograms of the simulated conditional posteriors of $\boldsymbol{\theta}_{*q}$ were placed in Figure 4.10.

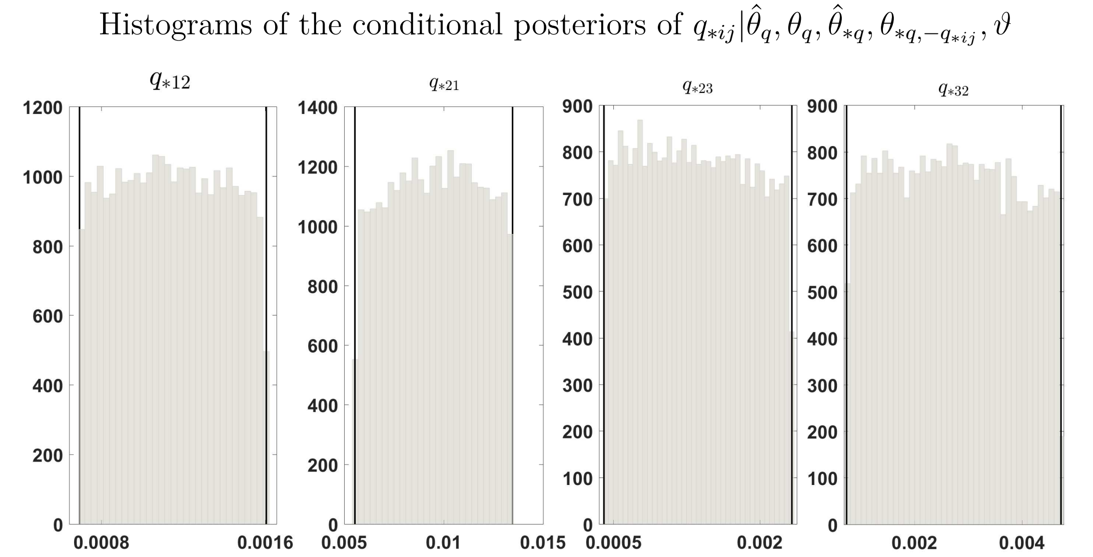


Figure 4.10: $q_{*ij}|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta$ when true $\boldsymbol{\theta}_{*q} \approx (0.0010, 0.0099, 0.0010, 0.0020)$ and $\hat{\boldsymbol{\theta}}_{*q} \approx (0.0010, 0.0090, 0.0008, 0.0022)$

From Figure 4.10, it was observed that the posterior of q_{*ij} did not reduce the bias or improve estimation of q_{*ij}. The posteriors look as if they could be random walks. Therefore, an alternative to the structure of (4.52) was considered. Another possible structure assumes that the noise was proportional to $\mu\left(\boldsymbol{\theta}_q\right)$. Therefore, $\hat{\boldsymbol{\theta}}_q$ was assumed to be the product of the unknown function of $\left(\mu\left(\boldsymbol{\theta}_q\right)\right)$ and the exponential function of white noise $(\boldsymbol{\varepsilon})$ in 4.87.

$$\hat{\boldsymbol{\theta}}_q = \mu\left(\boldsymbol{\theta}_q\right) e^{\varepsilon} \qquad\qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma_{\varepsilon}^2) \tag{4.87}$$

(4.87) cannot be modeled using (4.53). In (4.53), the variance of the noise is constant, but the noise in (4.87) was proportional to $\mu\left(\boldsymbol{\theta}\right)$. Therefore (4.53) was not an appropriate model for (4.87). In (4.88), (4.87) was considered when the data was $\left(\log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$.

$$\begin{aligned} \log\left(\hat{\boldsymbol{\theta}}_q\right) &= \log\left(\mu\left(\log\left(\boldsymbol{\theta}_q\right)\right) e^{\boldsymbol{\varepsilon}}\right) \\ \log\left(\hat{\boldsymbol{\theta}}_q\right) &= \log\left(\mu\left(\log\left(\boldsymbol{\theta}_q\right)\right)\right) + \boldsymbol{\varepsilon} \end{aligned} \tag{4.88}$$

(4.88) has constant noise and could be modeled with (4.53). It is important to note that the same could be done with the data $\left(\boldsymbol{\theta}_q, \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$, but this choice was made due to the belief that $\mu\left(\boldsymbol{\theta}_q\right)) \approx \boldsymbol{\theta}_q$. Depending on the data, $\left(\boldsymbol{\theta}_q, \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$ could be preferable.

Given this, the sampler was run on the $\left(\log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$ for the same dataset. The same algorithm described in (4.82) was used except $\log\left(\boldsymbol{\theta}_q\right)$ was used in place of all $\boldsymbol{\theta}_q$ and $\log\left(\hat{\boldsymbol{\theta}}_q\right)$ was used in place of all $\hat{\boldsymbol{\theta}}_q$. To compare the assumptions of the two proposed structures, posterior for $\boldsymbol{\theta}_q \beta_{q_{ij}} + \boldsymbol{f_{q_{ij}}}\left(\boldsymbol{\theta_q}\right) | \hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \vartheta$ were simulated for both (4.53) and (4.88). Since q_{23} and q_{32} were most problematic, quantile-quantile plots were placed in Figure 4.11 comparing the residual of the posteriors with the normal distribution.

It was observed in Figure 4.11 that the posteriors of $\boldsymbol{\theta}_q \beta_{q_{ij}} + \boldsymbol{f_{q_{ij}}}\left(\boldsymbol{\theta_q}\right) | \hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \vartheta$ generated using $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ had data that deviated quite significantly from a normal distribution for low values of q_{23} and q_{32}. The evidence does not support the assumption of normal noise with constant variance for (4.53). The quantile-quantile plots of the posteriors generated using $\left(\log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$ showed less deviation from the normal distribution. Therefore, posteriors for $\log\left(q_{*ij}\right) | \log\left(\hat{\boldsymbol{\theta}}_q\right)$, $\log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_{*q}\right), \log\left(\boldsymbol{\theta}_{*q,-q_{*ij}}\right), \vartheta_{\log}$ were simulated to test for improved performance. The posteriors of $\log\left(q_{*ij}\right)$ were not as , so they were transformed back to q_{*ij}. The transformations of the histograms were plotted in Figure 4.12 along with $\hat{q}_{*ij}$ when $\boldsymbol{z}_1^T$ was known, $\hat{q}_{*ij}$ from the algorithm in 4.2.2, and the mean of

Qunatile-Quantile Plots of the residuals of $\boldsymbol{\theta}_q\beta_{q_{ij}} + \boldsymbol{f}_{\boldsymbol{q_{ij}}}\left(\boldsymbol{\theta_q}\right)|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \vartheta$ and $\log\left(\boldsymbol{\theta}_q\right)\beta_{\log(q_{ij})} + f_{\log(q_{ij})}\left(\log\left(\boldsymbol{\theta}_q\right)\right)|\log\left(\hat{\boldsymbol{\theta}}_q\right), \log\left(\boldsymbol{\theta}_q\right), \vartheta_{\log}$ versus the normal distribution

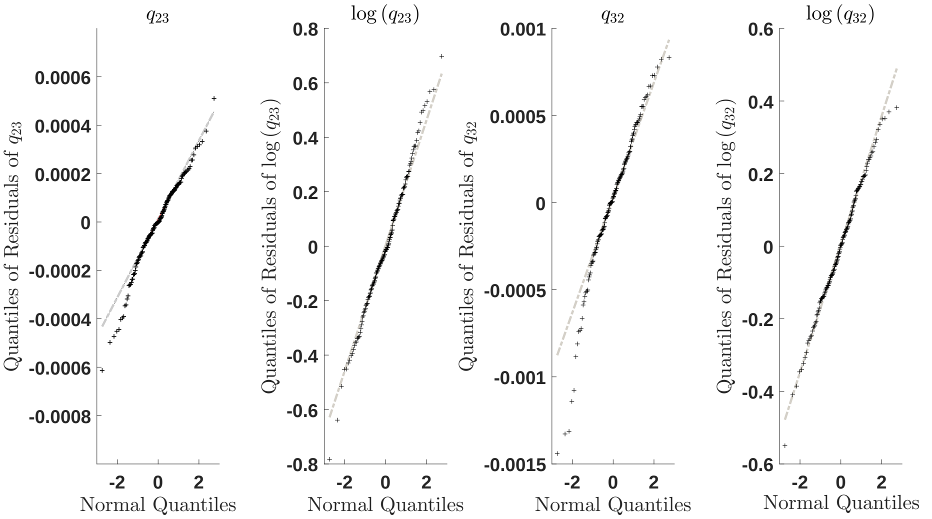


Figure 4.11: q_{*ij} denotes (4.53) was used and $\log(q_{*ij})$ denoted (4.88) was used.

the posterior.

The simulations of posterior distributions from (4.88) produced plausible results. The means of the posterior of q_{*12}, q_{*21}, and q_{*23} were closer to $\hat{q}_{*ij}$ when $\boldsymbol{z}_1^T$ was known than the estimation using the parameter training from Section 4.2.2. Although that was not the case for q_{*32}, the model suggested in (4.88) reduced the overall bias from $\hat{\boldsymbol{\theta}}_{*q}$. In addition, it was observed in Figure 4.9, that the bias of $\hat{r}_{32}$ and consequently $\hat{q}_{32}$ was small relative to the noise. Therefore, for the case or q_{*32} it was conceivable that $\hat{q}_{32}$ using the parameter training introduced in Section 4.2.2 could be closer to $\hat{q}_{*32}$ when $\boldsymbol{z}_1^T$ was given.

In addition, the estimates of $\boldsymbol{\beta}_{\log}$ acted as hypothesized. For each $\boldsymbol{\beta}_{\log(q_{ij})}$, the dimension that corresponded with q_{ij} had a mean near 1, all others were near 0. Due to the representation of the other parameters, the desired behaviors of those parameters were not as clear as $\boldsymbol{\beta}_{\log(q_{ij})}$. Following that, the posteriors of $\boldsymbol{l}^2_{\log(q_{ij})}$,

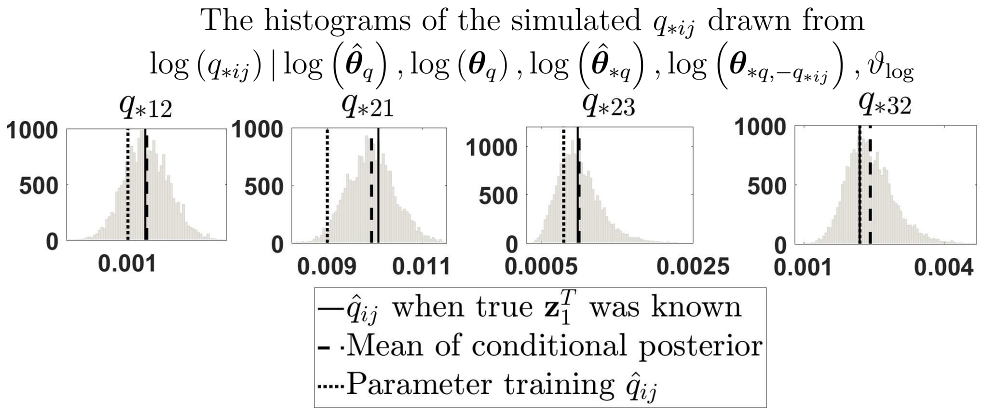


Figure 4.12: The transformed posterior from $\log\left(q_{*ij}\right) \mid \log\left(\hat{\boldsymbol{\theta}}_q\right), \log\left(\boldsymbol{\theta}_q\right)$ $, \log\left(\hat{\boldsymbol{\theta}}_{*q}\right), \log\left(\boldsymbol{\theta}_{*q,-q_{*ij}}\right), \vartheta_{\log}$

the posteriors of $\sigma^2_{\varepsilon_{\log(q_{ij})}}$, $\left(s^2_{\log(q_{ij})}\nu^2_{\log(q_{ij})}\right)$, and a list of the conditional priors can be found in appendix D.2.

Finally, the simulated values of $s^2_{\log(q_{ij})}$ provided confirmation of the use of the Gaussian process within the meta-model. $s^2_{\log(q_{ij})}$ can be thought of as the magnitude of the difference of the meta-model from the mean of the regression term. The histograms of the simulated posteriors of $s^2_{\log(q_{ij})}$ were placed in Figure 4.13.

In Figure 4.13, it was seen that the credible intervals of $s^2_{\log(q_{ij})}$ did not cover 0. Therefore, there was some non-linear relation between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$. Furthermore, since the meta-model was on the log scale, the scale of $s^2_{\log(q_{ij})}$ was approximately the relative difference from the linear regression term. By comparing that approximate to the relative errors observed in Figure 4.9, the magnitude of each $s^2_{\log(q_{ij})}$ was large enough that the Gaussian process within the meta-model was warranted.

The performance of the parameter training using the meta-model for bias reduction was tested two other ways. First, the Gibbs sampler introduced in the next section supplements the algorithm here and agreement between the two provided more confirmation of desired behaviors. Secondly, there were additional datasets that showed bias reduction of the parameter training estimates were effective. Therefore, this additional evidence was provided in a joint analysis of both the parameter training with bias correction and the Gibbs sampler.

The histograms of the simulated $s^2_{\log(q_{ij})}$ drawn from
$s^2_{\log(q_{ij})} | \log\left(\hat{\boldsymbol{\theta}}_q\right), \log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_{*q}\right), \log\left(\boldsymbol{\theta}_{*q,-q_{*ij}}\right), \vartheta_{\log - s^2_{\log\left(q_{ij}\right)}}$

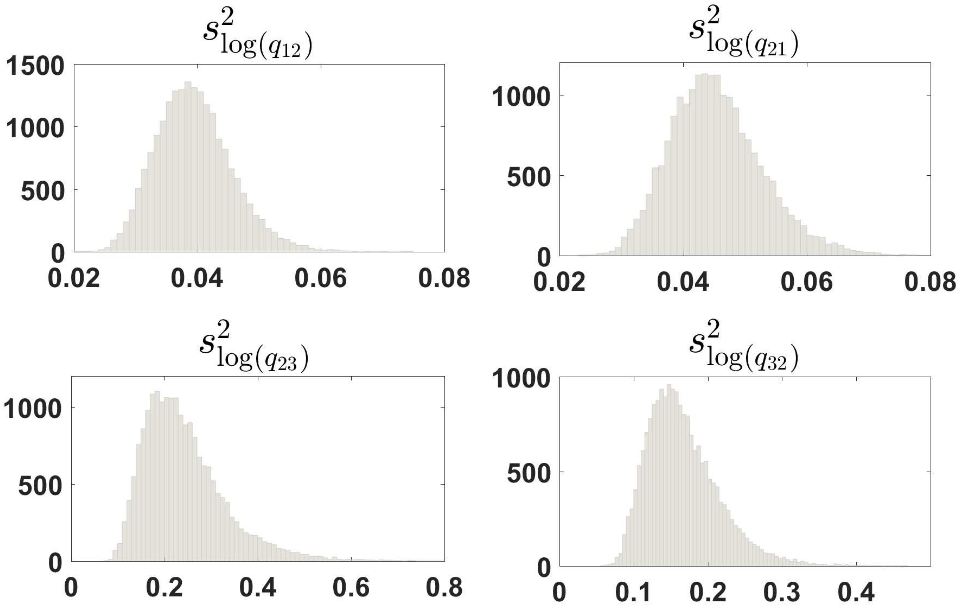


Figure 4.13: Histograms of "magnitude of the the non-linear part of meta-model" which on the log scale is approximately relative differences between the regression and full meta-model

Generating datasets

The goal of this research was to learn an unknown $\boldsymbol{\theta}_{*q}$ of a dataset. The parameter training algorithm was used to calculate a bias estimator $\hat{\boldsymbol{\theta}}_{*q}$. The bias was reduced by creating a meta-model for a computer generated set of $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs and solving the inverse problem. For quality inference, an appropriate set of $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs needed to identified. First, the center of the region for the pairs was chosen. This was done under the assumption that the ratio $\frac{\boldsymbol{\theta}}{\hat{\boldsymbol{\theta}}}$ was approximately constant in a small area around $\boldsymbol{\theta}_*$. Therefore, a guess for $\boldsymbol{\theta}_*$ was generated using the algorithm listed as (4.89) where n was the number of "guesses" desired.

Calculating an initial guess for $\boldsymbol{\theta}_{*q}$ (4.89)

```
I. Calculate $\hat{\boldsymbol{\theta}}_*$ for dataset with unknown $\boldsymbol{\theta}_*$.
II. $\boldsymbol{\theta}^{(1)} = \hat{\boldsymbol{\theta}}_*$
III. For $i = 1 : n$
    i. Simulate a dataset with parameters $\boldsymbol{\theta}^{(i)}$
    ii. Compute $\hat{\boldsymbol{\theta}}^{(i)}$ for the dataset in step i. using the parameter training algorithm developed for this research
    iii. $\boldsymbol{\theta}^{(i+1)} = \frac{\boldsymbol{\theta}^{(i)}}{\hat{\boldsymbol{\theta}}^{(i)}}$
IV. end
V. Let the initial guess $\boldsymbol{\theta}^{(g)}$ be $\boldsymbol{\theta}^{(i)}$ such that $\hat{\boldsymbol{\theta}}^{(i)}$ was closest to $\hat{\boldsymbol{\theta}}_*$
```

Then a set of $\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}\right)$ pairs was generated around $\boldsymbol{\theta}^{(g)}$. Initially, (4.89) was used to help determine the region to explore. However, the Gaussian process model used requires the region of exploration to be large relative to the area explored in (4.89). In fact, the restrictions of the parameter training algorithm discussed in Section 4.2.1 had the greatest effect on region size. For a 10 second dataset with the aforementioned parameters, the dimensions of q_{23} and q_{32} needed to be as big as possible without leaving the region where the parameter training algorithm was effective. If datapoints were evaluated outside the effective range of the parameter training algorithm, the associated $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs were often nonsensical and made learning ϑ difficult. This, in turn, made inference on $\boldsymbol{\theta}_{*q}$ poor. An example of inference where this occurred was placed in Figure 4.14(A).

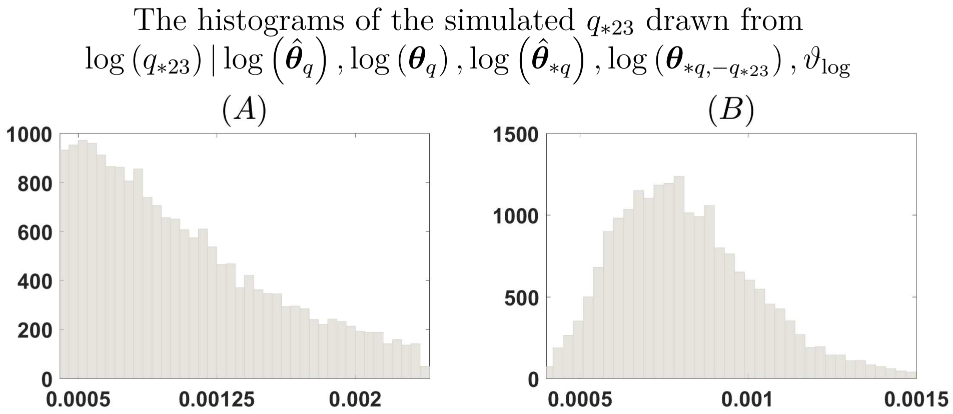


Figure 4.14: (A) shows a region that generated points outside the effective region of the parameter training and (B) has a region that was too restrictive to describe the posterior

Another difficulty that occurred was when the region was too small. In this case, the tails of the distribution of $\boldsymbol{\theta}_{*q}$ were not explored. An example of this was placed in Figure 4.14(B). Since the region in which the parameter training was effective was partially determined by the size of the dataset, this made inference difficult when the total time was too small. In the example from Figure 4.14(B), the tails of $\hat{q}_{*23}$ and $\hat{q}_{*32}$ were not completely explored given 10 seconds of data. With the correct region size, the tails of the distributions of q_{*ij} were explored (See Figure 4.12). However, finding the correct region size was not trivial for a dataset 10 seconds long. Therefore, it would be difficult to compute the bias reduction with much less than 10 seconds of data.

The method for generating an appropriate computer dataset was empirical. First, the center of the region was $\boldsymbol{\theta}^{(g)}$ which was calculated in (4.89). Then, the size of the region was decided using results as seen in Figure 4.14. Since Latin hypercubes are known to be a computationally cheap way to explore high dimensions [49], $\boldsymbol{\theta}_q$ for each ordered pair was generated by a Latin hypercube [36]. For this, the data was assumed to be a truncated normal distribution around

$\boldsymbol{\theta}^{(g)}$. The first reason for the normal assumption was that $\boldsymbol{\theta}^{(g)}$ tended to be a reasonable guess and therefore it was a good way to produce data concentrated near $\boldsymbol{\theta}_{*q}$. In addition, the normal distribution was chosen because the effective region of the parameter algorithm could not be defined as a single polytope. As seen in Section 4.2.1, the algorithm struggled when combinations of input variables made the parameters difficult to learn. If too many of the components of $\boldsymbol{\theta}_q$ were near the edge of the region proposed, a nonsensical $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pair could occur. By using the normal distribution, it was more likely that points near the edge of one dimension would be paired with points near the center of the other dimensions.

To generate the Latin hypercube with n datapoints, first, consider only one dimension. Divide the aforementioned dimension into n regions of equal probability. Within each of the n regions randomly generate one point (producing n coordinates in that dimension). This process was repeated for all other dimensions. Finally, each coordinate was randomly paired with coordinates from each of the other dimensions making n points within the Latin hypercube. In this case, there was no covariance applied between dimensions as using the normal distribution was sufficient. However, a Latin hypercube with a covariance structure is possible to generate, but a different implementation would be necessary. As an example, an explanation of the region used to generate the bias reduction from Section 4.2.6 can be found in Table 4.1.

Dimension	Center $\left(\boldsymbol{\theta}^{(g)}\right)$	Distribution of q_{ij}
q_{12}	0.0011	$\mathcal{TN}\left(0.0011, 0.1049^2, [0.0007, 0.0016]\right)$
q_{21}	0.0090	$\mathcal{TN}\left(0.0090, 0.2222^2, [0.0055, 0.0135]\right)$
q_{23}	0.0008	$\mathcal{TN}\left(0.0008, 0.3498^2, [0.0004, 0.0023]\right)$

Dimension	Center $\left(\boldsymbol{\theta}^{(g)}\right)$	Distribution of q_{ij}
q_{32}	0.0022	$\mathcal{TN}\left(0.0022, 0.3736^2, [0.0007, 0.0047]\right)$

Table 4.1: A description of the Latin hypercube used for the computer experiment in Section 4.2.6

The algorithm in (4.89) produced a reasonable $\boldsymbol{\theta}^{(g)}$ relatively quickly. Furthermore, (4.89) was expensive because it produced $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs sequentially. At the completion of (4.89), a Latin hypercube could be generated. With the Latin hypercube, generating $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs was embarrassingly parallel. Therefore, given enough processors, it was fairly cheap to generate a sufficient amount of $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs.

To test the precision of the bias reduction, the Gaussian process regression model was tuned to three different Latin hypercubes built for the dataset explored in 4.2.6. The histograms of the simulated posteriors of q_{*ij} were placed in Figure 4.15. Not surprisingly, the largest differences were observed in the posteriors of q_{*23} and q_{*32}. q_{*23} and q_{*32} both had long tails, and the majority of the difference was the size of those tails. This could have been cause by randomness in the generation of the dataset. As mentioned earlier, the inference near the edges of the region which the parameter training algorithm was effective was more difficult. If these edge points had too much randomness q_{*23} or q_{*32} could have come from an area near the edge. This is similar to the effect observed in Figure 4.14(A) but at a much lesser extent. This could be mitigated by having a longer time series, thus making the effective region of the algorithm larger. Alternatively, the edges of the region could be explored more thoroughly. Finally, adding more points to the Latin hypercube could be effective, but only if the region boundaries were well explored. To quantify the difference observed in these three posteriors, it was instructive to look at mean of each posterior. $\bar{q}_{*12}$ was 0.001 and $\bar{q}_{*21}$ was 0.0099 for all three Latin hypercubes. $\bar{q}_{*23}$ and $\bar{q}_{*32}$ were in the ranges $[0.00099, 0.001]$ and $[0.0023, 0.0024]$ respectively. The differences were small relative to the credible intervals and the relative bias introduced. Therefore, the performance could be improved by continuing to tune the region size while increasing the size of the Latin hypercube, but the uncertainty added here was small relative to the noise of the time series datasets.

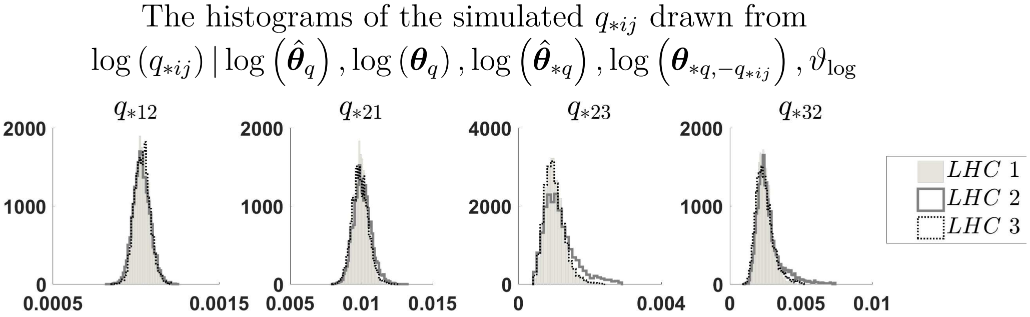


Figure 4.15: Each plot compared the histograms of three posteriors of q_{ij} simulated from three different Latin hypercubes

4.2.7 A Gibbs sampler for a 3 state system with only 2 distinct signals

likelihood

The traditional forward filtering backward sampling method was not an effective way to compute posteriors of the transition probabilities or rates for the three state system with only two signals. Therefore, the composite state was considered. By considering the composite state S_2, inference on the posterior distribution of $\boldsymbol{\theta} = (q_{12}, \boldsymbol{v}, \varphi_1, \varphi_2, w, \mu_{S_1}, \mu_{S_2}, \sigma^2)$ could be made. Since the additional likelihood required the consecutive observations in state S_2 for all stays in

S_2, Bayesian methods were convenient since the posterior produced a conditional posterior path or $\boldsymbol{z}_1^T$ for each draw. That, in turn, was used to generate the "dwell times" or consecutive observations in the composite state which was denoted as $\boldsymbol{d}$. In addition, this algorithm did not seem to exhibit the bias results as with the parameter training introduced in Section 4.2.2. Unfortunately, these advantages came at a cost, as simulating the full posteriors was very computationally expensive.

Like the parameter training from Section 4.2.2, the Gibbs sampler used two different likelihoods. The first was the likelihood for the traditional hidden Markov model. It was listed originally as (4.31) to (4.35), and was repeated here as (4.90) to (4.94).

$$p(\boldsymbol{y}_0^T, \boldsymbol{z}_0^T|\boldsymbol{\theta}) = p(\boldsymbol{z}_0|\boldsymbol{\theta}) \left[\prod_{i=1}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}, \boldsymbol{\theta})\right] \left[\prod_{i=0}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})\right] \tag{4.90}$$

$$p(\boldsymbol{z}_0|\boldsymbol{\theta}) = \rho_1^{z_{10}} \rho_2^{z_{20}} \rho_3^{z_{30}} \tag{4.91}$$

$$p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}, \boldsymbol{\theta}) = \boldsymbol{Q(\theta) z'_{i-1}} \tag{4.92}$$

$$p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \left[\mathcal{N}_{y_i}(\mu_{S_1}, \sigma^2)\right]^{z_{1i}} \left[\mathcal{N}_{y_i}(\mu_{S_2}, \sigma^2)\right]^{z_{2i}+z_{3i}} \tag{4.93}$$

$$\boldsymbol{Q(\theta)} = \begin{bmatrix} 1 - q_{12} & q_{21}(\boldsymbol{\theta}) & 0 \\ q_{12} & 1 - (q_{21}(\boldsymbol{\theta}) + q_{23}(\boldsymbol{\theta})) & q_{32}(\boldsymbol{\theta}) \\ 0 & q_{23}(\boldsymbol{\theta}) & 1 - q_{32}(\boldsymbol{\theta}) \end{bmatrix} \tag{4.94}$$

However, not all q_{ij} in the transition probability matrix were part of $\boldsymbol{\theta}$. $q_{21}(\boldsymbol{\theta})$, $q_{23}(\boldsymbol{\theta})$, and $q_{32}(\boldsymbol{\theta})$ were not parameters, but they were functions of parameters. They were listed originally list as (4.95) to (4.97) where there was more context and repeated as (4.7) to (4.9).

$$q_{21} = \varphi_2 + w(\varphi_1 - \varphi_2) \tag{4.95}$$

$$q_{23} = \frac{(\varphi_1 - \varphi_2)^2 - [(\varphi_1 - \varphi_2) - 2w(\varphi_1 - \varphi_2)]^2}{2[\varphi_2 + w(\varphi_1 - \varphi_2)]} \tag{4.96}$$

$$q_{32} = \varphi_2 - w(\varphi_1 - \varphi_2) - \frac{(\varphi_1 - \varphi_2)^2 - [(\varphi_1 - \varphi_2) - 2w(\varphi_1 - \varphi_2)]^2}{2[\varphi_2 + w(\varphi_1 - \varphi_2)]} \tag{4.97}$$

To make inference on the parameters that determine $q_{21}(\boldsymbol{\theta})$, $q_{23}(\boldsymbol{\theta})$, and $q_{32}(\boldsymbol{\theta})$ the dwell time in the composite state which was defined as $S_2 = S_{2A} \cup S_{2B}$ was used. The probability of the dwell time was listed as (4.98) and for more information on the composite state and the distribution of the dwell time see Section 4.1.2.

$$p(\boldsymbol{d}|\boldsymbol{\theta}) = \prod_{h=1}^{n} \left[w(1-\varphi_1)^{d_h - 1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h - 1}\varphi_2 \right] \tag{4.98}$$

The Gibbs Sampler

The Gibbs sampler first uses $\boldsymbol{\theta}$ to sample $\boldsymbol{z}_1^T$ using forward filter backward sample. With $\boldsymbol{z}_1^T$, the parameters $\boldsymbol{\rho}$, q_{12}, μ_{S_1}, μ_{S_2}, and σ^2 could be sampled using the methodology of the k state k signal system from Section 2.4.2. Finally, the dwell times in the composite state could be taken from $\boldsymbol{z}_1^T$. This was used for inference on φ_1, φ_2, and w. The sampler with N samples was listed as (4.99) and b_k indicates the k^{th} draw and $(b_j)_k$ indicated the same if b had a subscript.

The Gibbs sampler for 3 states with two discrete signals (4.99)

1. Pick $\boldsymbol{\theta}_0$

2. For $i = 1$ to N with $\boldsymbol{\theta}_{i-1}$

   (a) Draw z_T from 4.106

```
(b) for $l = T - 1$ to $l = 0$
        i. Draw $\boldsymbol{z}_l$ from 4.107
(c) Draw $(\boldsymbol{\rho})_i$ from 4.109
(d) Draw $(q_{12})_i$ from 4.111
(e) Draw $(\mu_{S_1})_i$ from 4.113
(f) Draw $(\mu_{S_2})_i$ from 4.115
(g) Draw $(\sigma^2)_i$ from 4.117
(h) Harvest $\boldsymbol{d}$ from $\boldsymbol{z}_0^T$
(i) for $h = 1$ to $H$
        i. draw $(v_h)_i$ from 4.119
(j) Draw $(\varphi_1)_i$ from 4.121
(k) Draw $(\varphi_2)_i$ from 4.122
(l) Draw $w_i$ from 4.124
```

Drawing $p(\boldsymbol{z}_1^T | \boldsymbol{y}_1^T, \boldsymbol{\theta})$

To draw from $\boldsymbol{z}_0^T$, forward filter backward sampling was used. The notations a_i, α_i, and β_i were used for brevity. Recall a_i, α_i, β_i were the state update, the state forecast, and the backward probabilities given $\boldsymbol{z}_{i+1}$ at time i. For a more detailed explanation see Section 2.4.2.

$$
\begin{aligned}
a_i(\boldsymbol{k}) &= p(\boldsymbol{z}_i = \boldsymbol{k} | \boldsymbol{y}_0^{i-1}, \boldsymbol{\theta}) \\
\alpha_i(\boldsymbol{k}) &= p(\boldsymbol{z}_i = \boldsymbol{k} | \boldsymbol{y}_0^{i}, \boldsymbol{\theta}) \\
\beta_i(\boldsymbol{k}) &= p(\boldsymbol{z}_i = \boldsymbol{k} | \boldsymbol{y}_0^{T}, \boldsymbol{\theta})
\end{aligned}
$$

First, a_i (the state update) and α_i were computed for $i = 0$ to T.

$$
\begin{aligned}
a_0(\boldsymbol{k}) &= p(\boldsymbol{z}_0 = \boldsymbol{k} | \boldsymbol{\theta}) \\
\alpha_0(\boldsymbol{k}) &= \frac{a_0(\boldsymbol{k}) N_{y_0}(\mu_{S_{\boldsymbol{k}}}, \sigma^2)}{\sum\limits_{\boldsymbol{j} \in \mathcal{Z}_0} a_0(\boldsymbol{j}) N_{y_0}(\mu_{S_{\boldsymbol{j}}}, \sigma^2)} \\
a_i(\boldsymbol{k}) &= \sum_{\boldsymbol{j} \in \mathcal{Z}_{i-1}} \alpha_{i-1}(\boldsymbol{j}) q_{\boldsymbol{jk}} \qquad (4.100) \\
\alpha_i(\boldsymbol{k}) &= \frac{a_i(\boldsymbol{k}) N_{y_i}(\mu_{S_{\boldsymbol{k}}}, \sigma^2)}{\sum\limits_{\boldsymbol{j} \in \mathcal{Z}_i} a_i(\boldsymbol{j}) N_{y_i}(\mu_{S_{\boldsymbol{j}}}, \sigma^2)} \qquad (4.101)
\end{aligned}
$$

Then the full conditionals were computed and $\boldsymbol{z}_i$ is sampled for $i = T$ to $i = 0$. (Note, for the case $i = T$, $\beta_T = \alpha_T$).

$$
\begin{aligned}
b_i(z_{i1} = 1) &= \alpha_i(z_{i1} = 1)(1 - q_{12}) z_{(i+1)1} + \alpha_i(z_{i1} = 1) q_{12} z_{(i+1)2} \qquad (4.102) \\
b_i(z_{i2} = 1) &= \alpha_i(z_{i2} = 1) \left(q_{21} z_{(i+1)1} + q_{22} z_{(i+1)2} + q_{23} z_{(i+1)3} \right) \qquad (4.103) \\
b_i(z_{i3} = 1) &= \alpha_i(z_{i3} = 1) q_{23} z_{(i+1)2} + \alpha_i(z_{i3} = 1)(1 - q_{32}) z_{(i+1)3} \qquad (4.104) \\
\beta_i(\boldsymbol{k}) &= \frac{b_i(\boldsymbol{k})}{b_i(z_{i1} = 1) + b_i(z_{i2} = 1) + b_i(z_{i3} = 1)} \qquad (4.105) \\
\boldsymbol{z}_T &\sim Dir_{\boldsymbol{z}_T} \left[\alpha_T(z_{T1} = 1), \alpha_T(z_{T2} = 1), \alpha_T(z_{T3} = 1) \right] \qquad (4.106) \\
\boldsymbol{z}_i &\sim Dir_{\boldsymbol{z}_i} \left[\beta_i(z_{i1} = 1), \beta_i(z_{i2} = 1), \beta_i(z_{i3} = 1) \right] \qquad i \neq T \qquad (4.107)
\end{aligned}
$$

Drawing $\boldsymbol{\rho}$, q_{12}, μ_{S_1}, μ_{S_2}, σ^2 from the posterior

$\boldsymbol{\rho}_k$, $(q_{12})_k$, $(\mu_{S_1})_k$,$(\mu_{S_2})_k$, and σ_k^2 were sampled using the likelihood functions in (4.90) to (4.94) and were discussed in more detail in Section 2.4.2. The conditional priors of $\boldsymbol{\rho}_k$, $(q_{12})_k$, $(\mu_{S_1})_k$,$(\mu_{S_2})_k$, and σ_k^2 were given as (4.108), (4.110), (4.112), (4.114), and (4.116) respectively. The posteriors were listed as (4.108), (4.111), (4.113), (4.115), and (4.117) respectively.

$$\boldsymbol{\rho} \sim Dir(\kappa_{\rho_1}, \kappa_{\rho_2}, \kappa_{\rho_3}) \tag{4.108}$$

$$\boldsymbol{\rho}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T \boldsymbol{\theta}_{-\boldsymbol{\rho}} \sim Dir_{\boldsymbol{\rho}}(z_{01} + \kappa_{\rho_1}, z_{02} + \kappa_{\rho_2}, z_{03} + \kappa_{\rho_3}) \tag{4.109}$$

$$q_{12} \sim Dir(\kappa_{q_{12}}, \kappa_{(1-q_{12})}) \tag{4.110}$$

$$q_{12}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T \boldsymbol{\theta}_{-q_{12}} \sim Dir\left(\sum_{i=1}^{T} z_{i2} z_{(i-1)1} + \kappa_{q_{12}}, \sum_{i=1}^{T} z_{i1} z_{(i-1)1} + \kappa_{(1-q_{12})}\right) \tag{4.111}$$

$$\mu_{S_1} = \mathcal{N}\left(m_{S_1}, \sigma_{S_1}^2\right) \tag{4.112}$$

$$\mu_{S_1}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T \boldsymbol{\theta}_{-\mu_{S_1}} \sim \mathcal{N}\left(\frac{\sigma_{S_1}^2 \sum\limits_{i=0}^{T} y_i z_{i1} + \sigma^2 m_{S_1}}{\sigma_{S_1}^2 \sum\limits_{i=0}^{T} z_{i1} + \sigma^2}, \frac{\sigma_{S_1}^2 \sigma^2}{\sigma_{S_1}^2 \sum\limits_{i=0}^{T} z_{i1} + \sigma^2}\right) \tag{4.113}$$

$$\mu_{S_2} \sim \mathcal{N}\left(m_{S_2}, \sigma_{S_2}^2\right) \tag{4.114}$$

$$\mu_{S_2}|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T \boldsymbol{\theta}_{-\mu_{S_2}} \sim \cdots \tag{4.115}$$

$$\cdots \mathcal{N}\left(\frac{\sigma_{S_2}^2 \sum\limits_{i=0}^{T} y_i (z_{i2} + z_{i3}) + \sigma^2 m_{S_2}}{\sigma_{S_2}^2 \sum\limits_{i=0}^{T} (z_{i2} + z_{i3}) + \sigma^2}, \frac{\sigma_{S_2}^2 \sigma^2}{\sigma_{S_2}^2 \sum\limits_{i=0}^{T} (z_{i2} + z_{i3}) + \sigma^2}\right)$$

$$\sigma^2 \sim \mathcal{IG}\left(\frac{n_0}{2}, \frac{d_0}{2}\right) \tag{4.116}$$

$$S_i = z_{i1}(y_i - \mu_{S_1})^2 + (z_{i2} + z_{i3})(y_i - \mu_{S_2})^2$$

$$\sigma^2|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T \boldsymbol{\theta}_{-\sigma^2} \sim \mathcal{IG}\left(\frac{n_0 + T + 1}{2}, \frac{d_0 + \sum_{i=0}^{T} S_i}{2}\right) \tag{4.117}$$

Drawing φ_1, φ_2, w, and v_h from the posterior

$\boldsymbol{z}_0^T$ determines the values for $\boldsymbol{d}$ so all necessary information to sample φ_1, φ_2, w from the posterior was available. Unfortunately, it is not easy to draw from the posterior of $p(d_h|\boldsymbol{\varphi}, w, v_h) = w(1-\varphi_1)^{d_h-1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h-1}\varphi_2$. Therefore, because it is computationally advantageous, an alternate form, as seen below, was used where v_h was introduced as a latent variable and indicates which geometric distribution d_h was drawn from.

$$\begin{aligned} p(d_h, v_h|\boldsymbol{\varphi}, w) &= \left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h} \left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h} \\ p(\boldsymbol{d}, \boldsymbol{v}|\boldsymbol{\varphi}, w) &= \prod_{h=1}^{m} \left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h} \left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h} \end{aligned}$$

The conditional priors of v_h, φ_1, φ_2, and w were listed as (4.118), (4.120), (4.120), and (4.123) respectively and the corresponding conditional posterior were listed as (4.119), (4.121), (4.122), and (4.124). $Bern(p)$ denotes a Bernoulli trial with probability of success p. A detailed explanation of a Gibbs sampler for a mixture of geometric distributions can be found in Section 2.2.2.

$$p(v_h|w, \varphi_1, \varphi_2) = p(v_h|w) = w^{v_h}(1-w)^{1-v_h} \tag{4.118}$$

$$v_h|d_h, \boldsymbol{\theta}_{-\boldsymbol{v_h}} \sim Bern\left(\frac{\left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]}{w(1-\varphi_1)^{d_h-1}\varphi_1 + (1-w)(1-\varphi_2)^{d_h-1}\varphi_2}\right) \tag{4.119}$$

$$\varphi_i \sim \mathcal{B}(\alpha_{\varphi_i}, \beta_{\varphi_i}) \tag{4.120}$$

$$\varphi_1 | \boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-\varphi_1} \sim \mathcal{B}\left(\sum_{h=1}^{H}(v_h) + \alpha_{\varphi_1}, \sum_{h=1}^{H}(v_h(d_h - 1)) + \beta_{\varphi_1}\right) \tag{4.121}$$

$$\varphi_2 | \boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-\varphi_2} \sim \mathcal{B}\left(\sum_{h=1}^{H}(1 - v_h) + \alpha_{\varphi_2}, \sum_{h=1}^{H}((1 - v_h)(d_h - 1)) + \beta_{\varphi_2}\right) \tag{4.122}$$

$$w \sim \mathcal{B}(\alpha_w, \beta_w) \tag{4.123}$$

$$w | \boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w} \sim \mathcal{B}\left(\sum_{h=1}^{H}(v_h) + \alpha_w, \sum_{h=1}^{H}(1 - v_h) + \beta_w\right) \tag{4.124}$$

Analysis of Gibbs sampler

Trial Name	True Values	MCMC Run 1	MCMC Run 2	MCMC Run 3
q_{12}	0.00099	0.0005	0.0080	0.0080
φ_1	0.0111	0.0080	0.0120	0.0200
φ_2	0.0018	0.0025	0.0018	0.0010
w	0.8711	0.7000	0.8500	0.9500
σ^2	9	12	8	4.5000
μ_{S_1}	32	33.6931	33.6931	33.6931
$\mu_{S_{2A}}$	26	29.0404	29.0404	29.0404

Table 4.2: Initial values for runs of the MCMC

The Gibbs sampler for this research was developed to make inference on the transition probabilities and subsequently the rates of the three state system with only two distinct signals. To analyze performance, the Gibbs sampler was run on the dataset from Section 4.2.2 with $1,000,000$ observations where $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$. First, the convergence of the algorithm was tested.

The convergence of the algorithm was tested using two methods. The first method was to check the draws 1 to 400 of an over-dispersed initial guess or $\boldsymbol{\theta}_0$. As with the two state, two signal system from Chapter 3, it was better to overestimate the difference of mixture parameters (μ_{S_1} and μ_{S_2} or φ_1 and φ_2) than underestimate. Therefore, using the same strategy from Chapter 3, $(\mu_{S_1})_0$ and $(\mu_{S_2})_0$ were picked to be the 75% and the 25% of $\boldsymbol{y}_1^T$ respectively. Unfortunately, a good proxy for that strategy did not exist for φ_1 and φ_2. The values for $\boldsymbol{\theta}_0$ were placed in Table 4.2. Figure 4.16 has the trace of the first four hundred draws from the posterior of $\boldsymbol{\theta}$ for Run 1, Run 2, and Run 3.

The traces of the first 400 draws from the conditional posteriors with over-dispersed $\boldsymbol{\theta}_0$

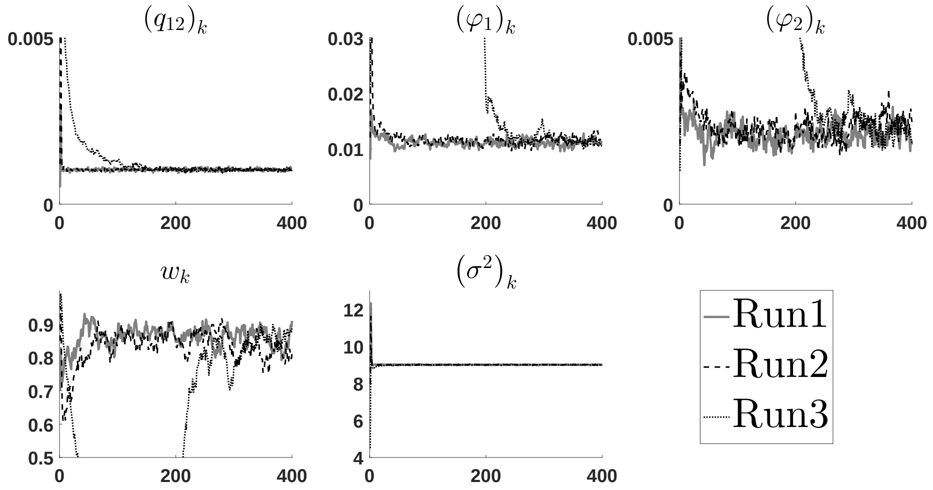


Figure 4.16: Each run was labeled as was in Table 4.2 where $\boldsymbol{\theta}_0$ was listed

In Figure 4.16 each trace, regardless of the quality of $\boldsymbol{\theta}_0$, finished in the same area. It was interesting to note that one choice of $\boldsymbol{\theta}_0$ caused the sampler to search an “undesirable area” of the posterior. Therefore, unlike the parameter training algorithm developed for this research, a bad $\boldsymbol{\theta}_0$ did not cause a failure. Since the Gibbs Sampler searches randomly, it should eventually return to the “correct region”, bad guesses just require more draws. Furthermore, once the sampler was in the ”correct region”, it stayed there. This behavior can be observed in the trace of draws 600-1000 in Figure 4.17.

The traces draws 600-1000 from the conditional posteriors with over-dispersed $\boldsymbol{\theta}_0$

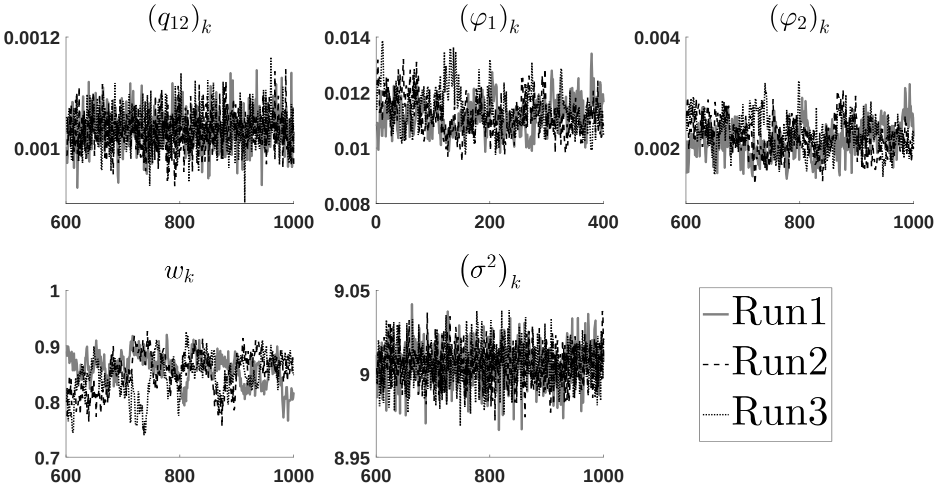


Figure 4.17: Each run was labeled as was in Table (4.2) where $\boldsymbol{\theta}_0$ was listed

From the trace of draws 600-1000, it can be seen that the three samples are all exploring the same general area that they converged to by the 400^{th} draw. However, it can also be seen that the draws of φ_1, φ_2, and w were highly correlated and the mixing was slow. Therefore, it was important that a large number of draws were taken from the posterior to produce satisfactory results.

As stated earlier, inference on three states with two distinct signals was difficult. Therefore, to make meaningful inference, a significant amount of information was required. To study the amount of data needed, the same dataset with $1,000,000$ observations where $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$ was used. The Gibbs sampler was run on the first second, the first two second, the first three seconds, ..., and all ten seconds.

Since q_{ij} and the resulting r_{ij} were the focus, the distributions of q_{ij} were studied. In particular, q_{23} and q_{32} were focused on as the inference on q_{23} and q_{32} was the most difficult. Figures 4.18 and 4.19 have histograms of simulated conditional posteriors of q_{23} and q_{32} for short datasets. For a dataset one second long, it can be seen that the algorithm explores much of the permissible range of q_{ij} and the concentration does not correlate with the true values in Table 4.2. At two seconds, the posterior has a concentration in the correct region, but the tails still explore a large area of the space. Finally, at 5 and 6 seconds, the tails dissipate some, and there was more concentration near the true value. However, the credible intervals were large enough that the error was more than an order of magnitude.

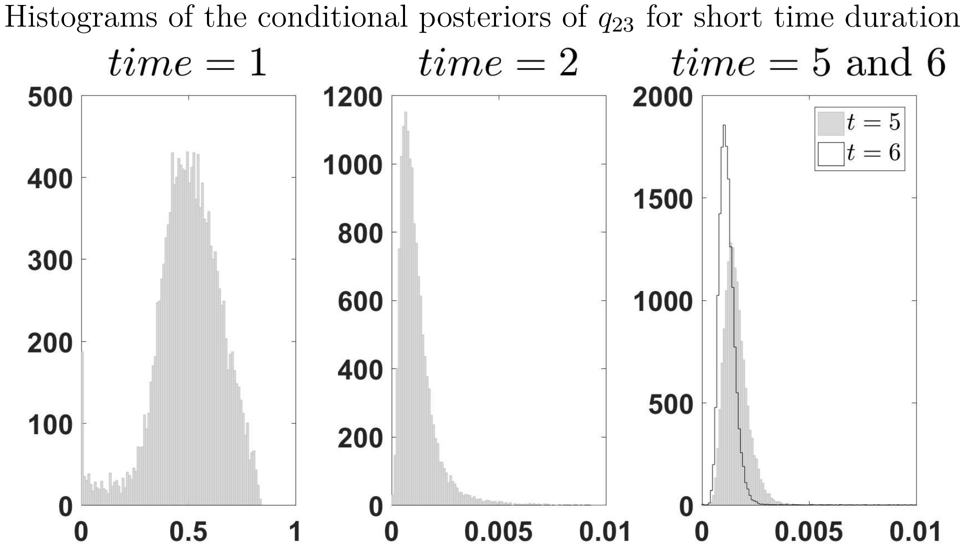


Figure 4.18: The datasets had a total time of 1, 2, 5, and 6 seconds

Although the dataset 6 seconds long produced credible intervals too big to be useful, it exhibited some desired behavior. Therefore, datasets of 7 or more seconds were examined more thoroughly. As before, the distributions for the posterior of parameter h were compared by using $\left(\epsilon_{h_{ij}}\right)_k$ where $\left(\epsilon_{h_{ij}}\right)_k + (h_{ij})_k = h_{ij}$. Then the 99%, 95%, and 80% credible intervals of set of $\left(\epsilon_{h_{ij}}\right)_k$ denoted as $\left\{\left(\epsilon_{h_{ij}}\right)_k\right\}$ were compared for different times using the demarcation described in Section 3.2.4. Figure 4.21 compared r_{ij} and Figure 4.20 compared μ_{S_1}, μ_{S_2} and σ^2. μ_{S_1}, μ_{S_2} and

Histograms of the conditional posteriors of q_{32} for short time duration

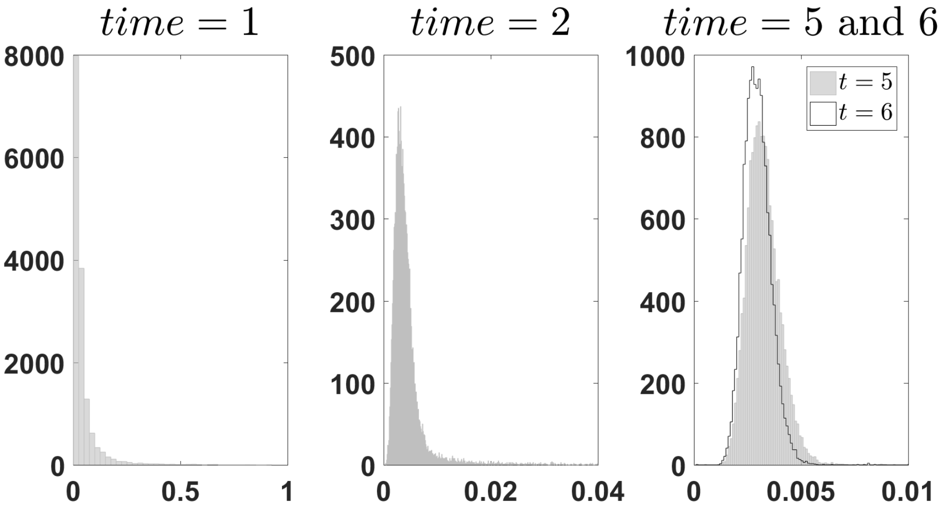


Figure 4.19: The datasets had a total time of 1, 2, 5, and 6 seconds

σ^2 showed very little relative error for all datasets from 7 to 10 seconds.

Histograms of the conditional posteriors of $\left\{\left(\epsilon_{h_{ij}}\right)_k\right\}$

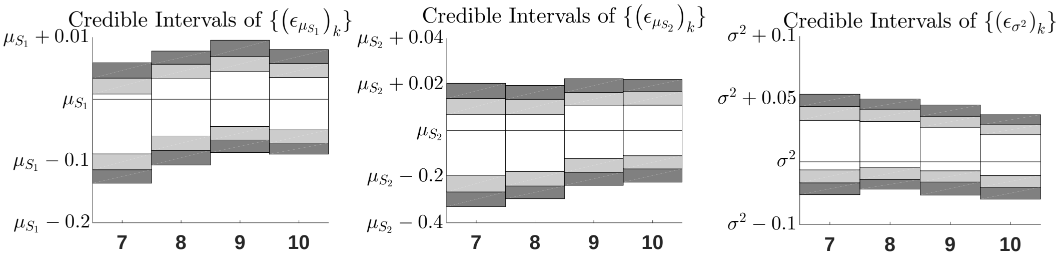
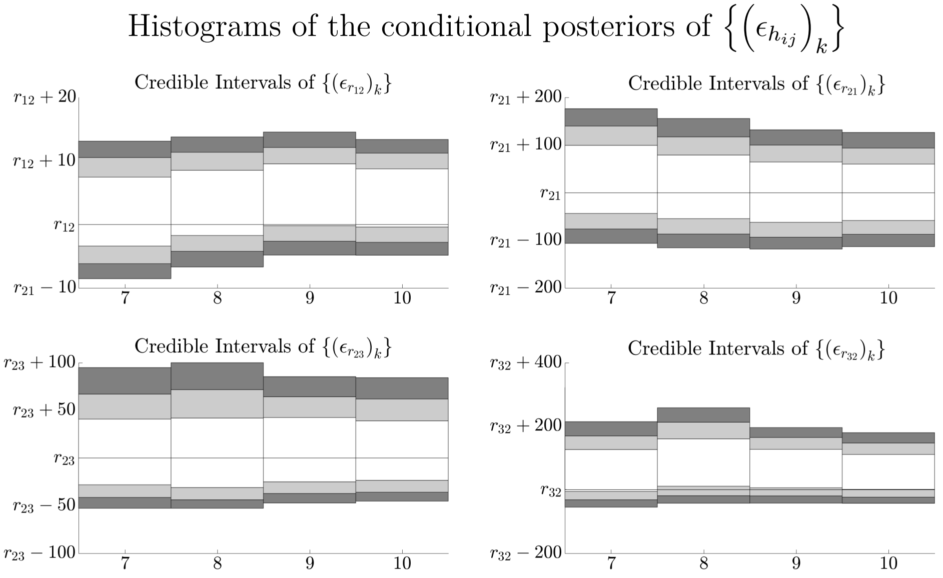


Figure 4.20: The total time of each dataset was labeled on the x axis

r_{12} showed reasonable credible intervals, and for 10 seconds the credible intervals were within the range of 10% relative error. r_{21} also exhibited positive traits and was within the range of 15% relative error for 10 seconds. r_{23} and r_{32} were more problematic. First, the true value of r_{32} was close to the edge of the credible intervals. However, since it was very close to the first border, it was just outside the 80% credible interval. It was not unrealistic for this to happen with one of the 7 parameters shown here. In addition, the credible intervals had ranges bigger for r_{23} and r_{32}. This was not surprising as there was no direct way to observe transitions between states S_{2A} and S_{2B}, but was problematic because running the Gibbs sampler for larger datasets was difficult. Running the Gibbs sampler on the 10 second dataset took just over a week. Finally, for more confirmation, the Gibbs sampler was run on more datasets and was compared to the parameter training with bias reduction. This was done in the following section.


Histograms of the conditional posteriors of $\left\{\left(\epsilon_{h_{ij}}\right)_k\right\}$

Credible Intervals of $\{(\epsilon_{r_{12}})_k\}$

Credible Intervals of $\{(\epsilon_{r_{21}})_k\}$

Credible Intervals of $\{(\epsilon_{r_{23}})_k\}$

Credible Intervals of $\{(\epsilon_{r_{32}})_k\}$

Figure 4.21: The total time of each dataset was labeled on the x axis

4.3 Comparison of methods

The goal of this research was to make inference on the transition rates of the three state system with two distinct signals. The parameter training developed for this research calculated point estimates for the rates. It was found that those estimates were biased and a meta-model was used to reduce that bias. The inference for the bias correction was Bayesian in nature and therefore simulated the posterior of the true rates.

The other method developed was a Gibbs sampler that sampled from the posteriors of the transition rates. This was convenient since both methods simulated posterior distribution of the transition rates and therefore were easy to compare. Both algorithms were run on the same dataset from 4.2.2 and 4.2.7 with $1,000,000$ observations where $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 100$, $r_{32} = 200$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$. Since the goal of this research was to compute transition rates, the posteriors of the rates or r_{ij} were used to compare the two algorithms. The posteriors of both were compared in Figure 4.22 where the posterior from the parameter training and bias reduction was denoted as r_{*ij} and the posterior simulated by the Gibbs sampler was denoted as r_{ij}. For reference, the maximum likelihood estimator of the transition rates given the true value of the state sequence of $\boldsymbol{z}_1^T$ was included on the graph.

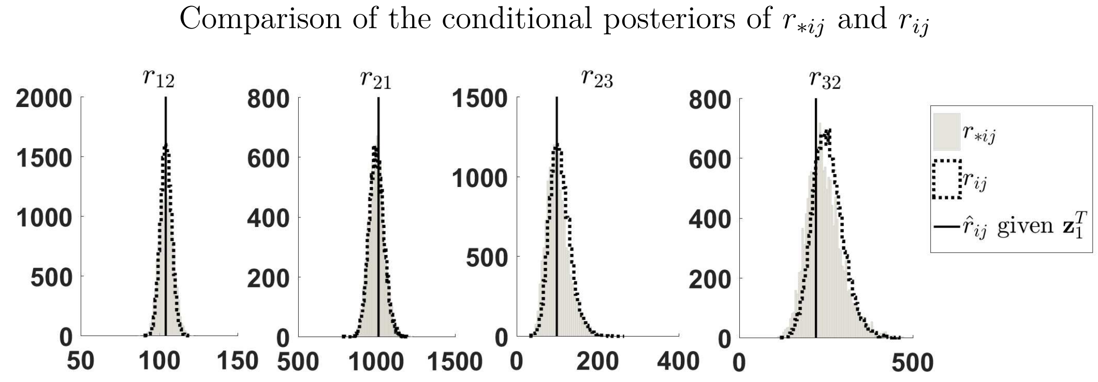


Figure 4.22: Histograms of posteriors of the two methods compared with the $\hat{r}_{ij}$ give the true $\boldsymbol{z}_1^T$

The inference of the Gibbs and the parameter training with bias reduction both seemed to centered on $\hat{r}_{ij}$ when the true $\boldsymbol{z}_1^T$ was known. In addition, the algorithms show a fair amount of agreement. The means of the posteriors from the Gibbs sampler for r_{12}, r_{21}, r_{23}, and r_{32} were 104.1594, 999.7081, 106.5893, and 254.6903. The corresponding means of the posteriors from the parameter training with bias reduction were 104.4262, 999.6526, 101.3179, and 244.9022 respectively. These were small differences relative to the size of the credible intervals.

Following this, a second time series was explored. The dataset was generated had $1,000,000$ observations with $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 200$, $r_{32} = 100$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, $\sigma^2 = 9$, and $\Delta t = \frac{1}{100000}$. Again, the histograms of the posteriors were compared.

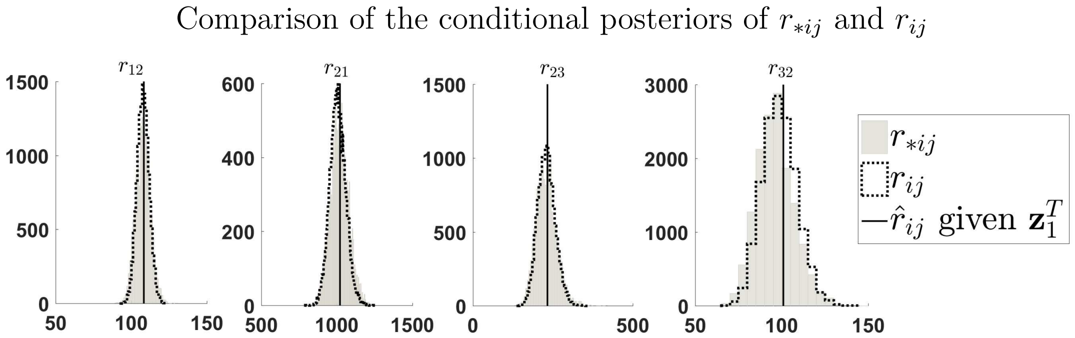


Figure 4.23: Histograms of posteriors of the two methods compared with the $\hat{r}_{ij}$ give the true $\boldsymbol{z}_1^T$

Figure 4.23 exhibited many of the same behaviors. First, the posteriors of the Gibbs sampler and the parameter training with bias reduction seemed centered on $\hat{r}_{ij}$ when $\boldsymbol{z}_1^T$ was known. Second, the difference between the two simulated posteriors was very small relative to the credible intervals. Finally, it can be observed in both Figure 4.22 and 4.23 that the credible intervals of the parameter training with bias reduction

Dataset Name	r_{12}	r_{21}	r_{23}	r_{32}
1	100	1000	100	200
2	100	1000	200	100
3	100	1000	100	100
4	100	1000	200	200

Table 4.3: List of Datasets

were slightly bigger. This comparison was done on four datasets, and this result was consistent. A list of the four experiments can be found in Table 4.3 where each dataset was 10 seconds long sampled at 100 kHz and $\mu_{S_1} = 32$, $\mu_{S_2} = 26$ and $\sigma^2 = 9$. The results were compared in Figure 4.24 using the credible intervals of $\left\{\left(\epsilon_{r_{ij}}\right)_k\right\}$. Each dataset was labeled on the x axis where a denotes the results from the Gibbs sampler and a_* denoted the results of the parameter training with bias reduction. For reference, the mean of each posterior was denoted as a dotted line. As stated previously, datasets 3 and 4 exhibited traits similar to datasets 1 and 2 where both distributions were very close, and the difference relative to the size of the credible intervals were small. Furthermore, in all four cases, the credible intervals of the parameter training with bias reduction were slightly bigger.

Comparison of the credible intervals of the conditional posteriors of $\left\{\left(\epsilon_{r_{*ij}}\right)_k\right\}$ and $\left\{\left(\epsilon_{r_{ij}}\right)_k\right\}$

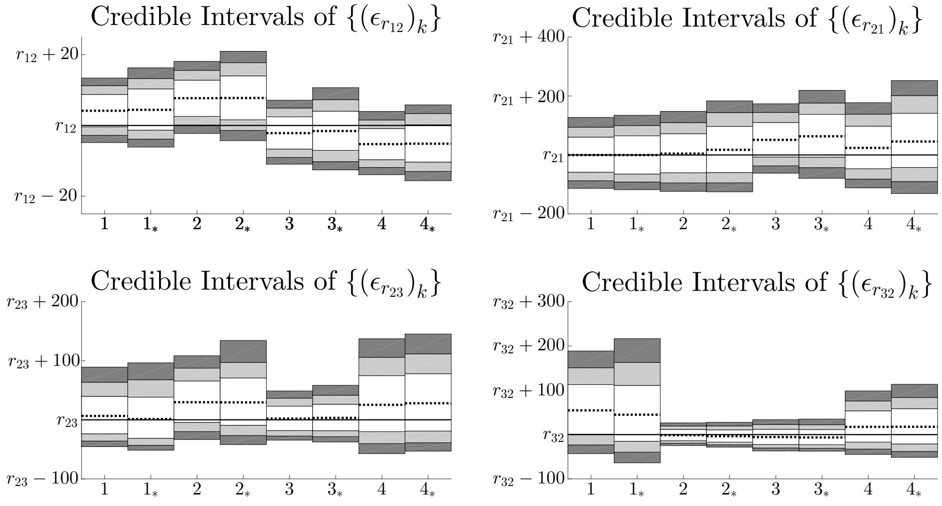


Figure 4.24: Comparison of the credible intervals of the conditional posteriors where true r_{ij} was represented by a solid line and the mean of the posteriors were represented by a dotted line

The agreement of the two methods provides additional justification in addition to the results previously discussed. The Gibbs sampler was a rigorous Bayesian statistical model of the biological system. It had slightly more concise credible intervals but required long computation times. For reference, the 10 second dataset with data collected at a frequency of 100 kHz took a little over a week to compute. The parameter training with bias reduction was an ad-hoc method for computing the posteriors. It required producing a set of $\left(\boldsymbol{\theta_q}, \hat{\boldsymbol{\theta}}_q\right)$ pairs to compute the posterior of $\boldsymbol{\theta}_{*q}$. Calibrating a set that was big enough to include the full range of the distributions of q_{ij} (and subsequently r_{ij}) without producing "bad $\left(\boldsymbol{\theta_q}, \hat{\boldsymbol{\theta}}_q\right)$ pairs" was done empirically and could be time consuming. However, computing the initial values of $\hat{\boldsymbol{\theta}}_q$ for a single dataset 10 seconds long at 100 kHz ranged from 5-10 minutes compute time. The Latin hypercube could be computed entirely in parallel if enough processors were available. Finally, running the bias reduction for 163 $\left(\boldsymbol{\theta_q}, \hat{\boldsymbol{\theta}}_q\right)$ pairs took under 1 hour. It is important to note, all codes were written in MATLAB and had not been optimized for speed. Times for both methods could be improved, but the parameter training with bias reduction was run on a dataset with 80 seconds of data when $\Delta t = \frac{1}{100000}$ in a half day.

Given that the parameter training with bias reduction could be run on larger datasets, these datasets were used as further confirmation of the methodology. As before, the credible intervals of $\left\{\left(\epsilon_{r_{ij}}\right)_k\right\}$ were placed on a single plot for datasets with total times of 10, 20, 40 and 80 seconds. Each dataset was collected at a frequency 100 kHz and the parameters of each dataset were $r_{12} = 100$, $r_{21} = 1000$, $r_{23} = 200$, $r_{32} = 100$, $\mu_{S_1} = 32$, $\mu_{S_2} = 26$, and $\sigma^2 = 9$. The credible intervals were placed in Figure 4.25 and the times were placed on the x-axis.

Figure 4.25 shows additional empirical evidence of desired behaviors. As the amount of data increases, the credible intervals shrink around the true values of r_{ij}. In addition, it was interesting to note that given 80 seconds of data, the

Comparison of the credible intervals of the conditional posteriors of $\left\{\left(\epsilon_{r_{*ij}}\right)_k\right\}$ for datasets from 10-80 seconds long

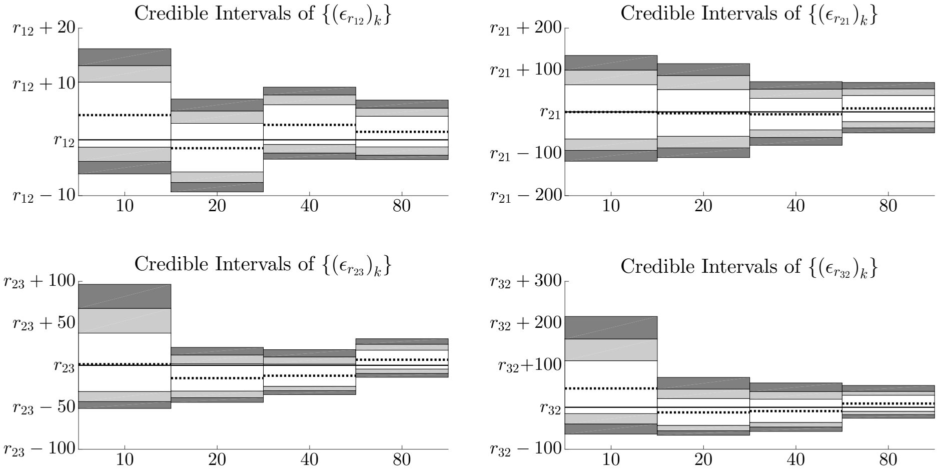


Figure 4.25: Comparison of the credible intervals of the conditional posteriors where true r_{ij} was represented by a solid line and the mean of the posteriors were represented by a dotted line

range of the 99% credible interval of the r_{23} was about $[86, 152]$ and r_{32} was about $[174, 252]$. Therefore, 99% credible intervals of the posteriors of r_{23} and r_{32} did not exhibit more than 0.33 and 0.27 relative error. Given that inference was being made on three states when only two states where visible, this result shows that the method can be precise with enough data.

4.4 Future Work

The nanopore experiments conducted by Lieberman and her colleagues resulted in a system with three states and two signals. The methods developed in this thesis provide a statistical model for the transition rates that supplement adhoc maximum likelihood approach based on dwell times employed by Lieberman et al [32]. Both methods developed here incorporated hidden Markov modeling to calculate the transition rates. The Gibbs sampler developed was a rigorous statistical model for generating posteriors of the of the transition rates. The parameter training with bias reduction was a more ad-hoc approach to the problem. It produced reasonable but slightly less precise credible intervals in much less time.

In the experiments modeled, the DNA was engineered not to go passed the post-translocation, ternary structure state. For a better understanding of the replication process, it would be necessary to understand all transition rates within DNA replication. To study all transitions, it is necessary to extend the models to fit DNA replication where the states were not artificially restricted. Following that, the biochemical states of DNA replication are not fully understood, and not all states can be observed. Building a test to determine whether "hidden" or composite states exist that were not previously known would be beneficial.

Computer experiments such as those introduced by Kennedy and O'Hagan are widely implemented. Using the experiment to address bias, discrepancy, or what they refer to model inadequacy was not a new application [27]. However, continual advances in computing technology have made collecting large datasets increasingly feasible and subsequently increased the demand for algorithms that can process that data in a reasonable time. Applying Gaussian process regression to "fast" but inadequate models may have a broader range of applications to be explored. Furthermore, the application of the Gaussian process regression resulted in a reasonable simulated posterior. Using a set of maximum likelihood estimators might be a fast way to estimate the true posterior of the model. Applying this in other cases could provide evidence for a viable alternative method to estimate credible intervals. In addition, if posteriors were not desired, using methods that made point estimates for learning the Gaussian process regression parameters such as described in [7] would be much faster.

The nanopore easily created large amounts of data. Furthermore, both methods required a lot of data to make inference with desirable credible interval sizes. Therefore, writing the code in a language other than MATLAB and optimizing would be needed if one wanted to evaluate large amounts of data (time series data with 8 million points were run using the current code).

As stated above, creating a Latin hypercube that was big enough so the full posterior could be explored without creating "bad" $\left(\boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q\right)$ pairs was difficult. Furthermore, it was seen in Section 4.2.6, that the Latin hypercube still contributed to small variability in the posteriors. Creating a better understanding of the theoretical limitations of inference on r_{ij} for the parameter training would help reduce computation time and produce more reliable posteriors.

Appendix A

Supplement: A mixture of two geometric distributions

A.1 The conditional posterior distributions of a mixture of two geometric distributions

A.1.1 Derivation of the conditional posterior of w for a mixture of two geometric distributions

$$\begin{aligned}
w &\sim \mathcal{B}(\alpha_w, \beta_w) \\
p(w|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w}) &\propto p(\boldsymbol{d}, \boldsymbol{v}|\varphi_1, \varphi_2, w)p(w) \\
p(w|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w}) &\propto \prod_{h=1}^{H} \left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h} \left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h} p(w) \\
p(w|\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{\theta}_{-w}) &\propto \prod_{h=1}^{H} [w]^{v_h} \left[(1-w)\right]^{1-v_h} w^{\alpha_w-1}(1-w)^{\beta_w-1}
\end{aligned}$$

$$p(w|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-w}) \propto w^{\sum\limits_{h=1}^{H}(v_h)}(1-w)^{\sum\limits_{h=1}^{H}(1-v_h)} w^{\alpha_w-1}(1-w)^{\beta_w-1}$$

$$p(w|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-w}) \propto w^{\sum\limits_{h=1}^{H}(v_h)+\alpha_w-1}(1-w)^{\sum\limits_{h=1}^{H}(1-v_h)+\beta_w-1}$$

$$w|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-w} \sim \mathcal{B}_w\left(\sum_{h=1}^{H}(v_h)+\alpha_w, \sum_{h=1}^{H}(1-v_h)+\beta_w\right)$$

A.1.2 Derivation of the conditional posteriors of φ_1 and φ_2 for a mixture of two geometric distributions

$$\varphi_1 \sim \mathcal{B}(\alpha_{\varphi_1},\beta_{\varphi_1})$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto p(\boldsymbol{d},\boldsymbol{v}|\varphi_1,\varphi_2,w)p(\varphi_1)$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto p(\boldsymbol{d},\boldsymbol{v}|\boldsymbol{\varphi},\varphi_2,w)p(\varphi_1)$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto \prod_{h=1}^{m}\left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h}\left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h} p(\varphi_1)$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto \prod_{h=1}^{m}\left[(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h}\varphi_1^{\alpha_{\varphi_1}-1}(1-\varphi_1)^{\beta_{\varphi_1}-1}$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto \varphi_1^{\sum\limits_{h=1}^{m}(v_h)}(1-\varphi_1)^{\sum\limits_{h=1}^{m}(v_h(d_h-1))}\varphi_1^{\alpha_{\varphi_1}-1}(1-\varphi_1)^{\beta_{\varphi_1}-1}$$

$$p(\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1}) \propto \varphi_1^{\sum\limits_{h=1}^{m}(v_h)+\alpha_{\varphi_1}-1}(1-\varphi_1)^{\sum\limits_{h=1}^{m}(v_h(d_h-1))+\beta_{\varphi_1}-1}$$

$$\varphi_1|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_1} \sim \mathcal{B}\left(\sum_{h=1}^{m}(v_h)+\alpha_{\varphi_1}, \sum_{h=1}^{m}(v_h(d_h-1))+\beta_{\varphi_1}\right)$$

The derivation of φ_2 was almost identical to φ_1 and the posterior was listed below.

$$\varphi_2|\boldsymbol{d},\boldsymbol{v},\boldsymbol{\theta}_{-\varphi_2} \sim \mathcal{B}\left(\sum_{h=1}^{m}(1-v_h)+\alpha_{\varphi_2}, \sum_{h=1}^{m}((1-v_h)(d_h-1))+\beta_{\varphi_2}\right)$$

A.1.3 Derivation of thec onditional posterior of v_h for a mixture of two geometric distributions

$$
\begin{aligned}
v_h|w &\sim Bernoulli(w)\\
p(v_h|\boldsymbol{d},\boldsymbol{\theta}) &\propto p(\boldsymbol{d}|\varphi_1,\varphi_2,w,v_h)p(v_h|w)\\
p(v_h|\boldsymbol{d},\boldsymbol{\theta}) &\propto \left[(1-\varphi_1)^{(d_h-1)}\varphi_1\right]^{v_h}\left[(1-\varphi_2)^{(d_h-1)}\varphi_2\right]^{1-v_h}\left[w^{v_h}(1-w)^{1-v_h}\right]\\
p(v_h|\boldsymbol{d},\boldsymbol{\theta}) &\propto \left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h}\left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h}\\
p(v_h|\boldsymbol{d},\boldsymbol{\theta}) &\propto \frac{\left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]^{v_h}\left[(1-w)(1-\varphi_2)^{d_h-1}\varphi_2\right]^{1-v_h}}{w(1-\varphi_1)^{d_h-1}\varphi_1+(1-w)(1-\varphi_2)^{d_h-1}\varphi_2}\\
v_h|\boldsymbol{d},\boldsymbol{\theta} &\sim Bernoulli\left(\frac{\left[w(1-\varphi_1)^{d_h-1}\varphi_1\right]}{w(1-\varphi_1)^{d_h-1}\varphi_1+(1-w)(1-\varphi_2)^{d_h-1}\varphi_2}\right)
\end{aligned}
$$

Appendix B

Supplement: HMMs and the Orstein-Uhlenbeck Process

B.1 Defining k and σ_η in terms of B and γ

The zero mean Orstein-Uhlenbeck Process

$$dX = -BXdt + \gamma dW$$

Properties of Wiener Process (W)

I $W(0) = 0$

II For $t_1 \leq t_2$, $W(t_2) - W(t_1) \sim N(0, t_2 - t_1)$

III For $t_1 \leq t_2 \leq t_3 \leq t_4$, $W(t_2) - W(t_1)$ and $W(t_4) - W(t_3)$ are independent.

A Dynamic Linear Model (DLM)

$$
\begin{aligned}
x_i &= X(t_i) \\
\Delta t &= t_i - t_{i-1} \; \forall \; i \in \{2, 3, ..., T\} \\
x_1 &\sim N(\mu_0, \sigma_\eta^2) \\
x_i &= kx_{i-1} + \eta_i & \eta_i \sim N(0, \sigma_\eta^2) \\
y_i &= x_i + \varsigma & \varsigma \sim N(0, \sigma_\varsigma^2)
\end{aligned}
$$

B.1.1 Deriving k in terms of B and γ

For ease all definitions were re-listed in appendix B.1. To solve for k, consider $dX = -BXdt + \gamma dW$.

$$
\begin{aligned}
dX &= -BXdt + \gamma dW \\
dX + BXdt &= \gamma dW \\
e^{Bt}dX + e^{Bt}BXdt &= e^{Bt}\gamma dW \\
\left(e^{Bt}X\right)_t &= e^{Bt}\gamma dW
\end{aligned}
$$

Since $t_i - t_{i-1} = \Delta t$, k can be solved for using $X(t_i)$ and $X(t_{i-1})$ without any loss of generality.

$$
\begin{aligned}
\left(e^{Bt}x\right)_t &= e^{Bt}\gamma dW \\
\int_{t_{i-1}}^{t_i} \left(e^{Bs}X\right)_s ds &= \int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) \\
\int_{t_{i-1}}^{t_i} \left(e^{Bs}X\right)_s ds &= \int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) \\
e^{Bt_i}X(t_i) - e^{Bt_{i-1}}X(t_{i-1}) &= \int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) \qquad \text{(B.1)}
\end{aligned}
$$

Unfortunately, the solution is a distribution and not deterministic. Consequently, $\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)$ is not integrable. However, the value of the first and second moments can be calculated. To compute these expectations, consider $\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)$ as a Riemann integral. For this, let $\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) = \lim_{N\to\infty} \sum_{j=1}^{N} \gamma e^{Bs_j} dW_j$ where s_1, s_2,..., s_j,..., s_{N+1} is a partition of $[t_{i-1}, t_i]$ and $dW_j = W(s_{j+1}) - W(s_j)$.

$$\begin{aligned}
\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) &= \lim_{N\to\infty} \sum_{j=1}^{N} \gamma e^{Bs_j} dW_j \\
\mathbb{E}\left[\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right] &= \mathbb{E}\left[\lim_{N\to\infty} \sum_{j=1}^{N} \gamma e^{Bs_j} dW_j\right] \\
\mathbb{E}\left[\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right] &= \lim_{N\to\infty} \sum_{j=1}^{N} \gamma e^{Bs_j} \mathbb{E}\left[dW_j\right]
\end{aligned}$$

From the properties of W, $dW_j = W(s_{j+1}) - W(s_j) \sim \mathcal{N}(0, s_{j+1} - s_j)$, Therefore, $\mathbb{E}\left[dW_j\right]] = 0 \ \forall \ j$. Following this, $\mathbb{E}\left[\int_0^t e^{Bs}\gamma dW(s)\right] = 0$. Using this fact and (B.1), the $\mathbb{E}\left[X(t)\right]$ given $X(t_{i-1})$ can be found.

$$\begin{aligned}
\mathbb{E}\left[e^{Bt_i}X(t_i) - e^{Bt_{i-1}}X(t_{i-1})\right] &= \mathbb{E}\left[\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right] \\
e^{Bt_i}\mathbb{E}\left[X(t_i)\right] - e^{Bt_{i-1}}X(0) &= 0 \\
\mathbb{E}\left[X(t_i)\right] &= e^{-Bt}e^{Bt_{i-1}}X(0) \\
\mathbb{E}\left[X(t_i)\right] &= e^{-Bt+Bt_{i-1}}X(0) \\
\mathbb{E}\left[X(t_i)\right] &= e^{-B(\Delta t)}X(0)
\end{aligned}$$

Matching notations from the continuous system to the discrete system, $\mathbb{E}[x_t] = kx_{t-1} = kX(t_{i-1}) = e^{-B(\Delta t)}X(t_{i-1})$ or $k = e^{-B(\Delta t)}$.

B.1.2 Deriving σ_η^2 in terms of B and γ

To solve for σ_η^2 in terms of B and γ, consider the second moment. To find $\mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right]$, again a Riemann integral was used.

$$\mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] = \lim_{N\to\infty}\mathbb{E}\left[\left(\sum_{j=1}^{N}\gamma e^{Bs_j}dW_j\right)^2\right]$$
$$\mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] = \lim_{N\to\infty}\mathbb{E}\left[\left(\sum_{j=1}^{N}\gamma e^{Bs_j}dW_j\right)\left(\sum_{k=1}^{N}\gamma e^{Bs_k}dW_k\right)\right]$$

In the case where $j \neq k$, each term has the form $\mathbb{E}\left[2\gamma e^{Bs_j}dW_j\gamma e^{Bs_k}dW_k\right]$ which is equal to $2\gamma^2 e^{B^2 s_j s_k}\mathbb{E}\left[dW_j dW_k\right]$. Since the s_i's formed a partition, then dW_j and dW_k are disjoint and therefore independent by the properties of the Wiener Process. Following this, when $j \neq k$ then $\mathbb{E}\left[2\gamma e^{Bs_j}dW_j\gamma e^{Bs_k}dW_k\right]$ is equal to $2\gamma^2 e^{B^2 s_j s_k}\mathbb{E}\left[dW_j\right]\mathbb{E}\left[dW_k\right]$ resulting in $2\gamma^2 e^{B^2 s_j s_k}\times 0\times 0 = 0$. This results in $\mathbb{E}\left[\left(\int_0^t e^{Bs}\gamma dW(s)\right)^2\right]$ being equal to $\lim_{N\to\infty}\mathbb{E}\left[\sum_{j=1}^{N}\gamma^2 e^{2Bs_j}dW_j^2\right]$ since it is only necessary to deal with the case $j = k$. Then $\lim_{N\to\infty}\mathbb{E}\left[\sum_{j=1}^{N}\gamma^2 e^{2Bs_j}dW_j^2\right]$ is equal to $\lim_{N\to\infty}\gamma^2\sum_{j=1}^{N}e^{2Bs_j}\mathbb{E}\left[dW_j^2\right]$. Therefore, it was necessary to find $\mathbb{E}[dW^2]$, which was done by considering $\text{Var}[dW]$.

$$\text{Var}(dW) = \mathbb{E}[dW^2] - \mathbb{E}[dW]^2$$
$$\text{Var}(W(t+dt) - W(t)) = \mathbb{E}[dW^2] - \mathbb{E}[dW]^2$$

From the properties of the Wiener Process it is known that the $Var(W(t + dt) - W(t)) = t + dt - t = dt$. Furthermore, in appendix B.1.1 it was found that

$E[dW] = 0.$

$$\begin{aligned} dt &= \mathbb{E}[dW^2] - (0)^2 \\ dt &= \mathbb{E}[dW^2] \end{aligned}$$

Using $\text{Var}(dW) = dt$, $\mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] = \lim_{N\to\infty} \gamma^2 \sum_{j=1}^{N} e^{2Bs_j}(s_{j+1} - s_j)$.

$$\begin{aligned} \mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] &= \lim_{N\to\infty} \gamma^2 \sum_{j=1}^{N} e^{2Bs_j}(s_{j+1} - s_j) \\ \mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] &= \int_{t_{i-1}}^{t_i} \gamma^2 e^{2Bs} ds \\ \mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] &= \gamma^2 \left[\frac{1}{2B} e^{2Bs}\right]_{t_{i-1}}^{t_i} \\ \mathbb{E}\left[\left(\int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s)\right)^2\right] &= \frac{\gamma^2}{2B}\left[e^{2Bt_i} - e^{2Bt_{i-1}}\right] \end{aligned}$$

Finally, above it was shown that $\text{Var}(dW) = \mathbb{E}[dW^2]$. Therefore, $\text{Var}(dW) = \frac{\gamma^2}{2Bs}\left[e^{2Bt_i} - e^{2Bt_{i-1}}\right]$. Consider this fact with (B.1).

$$\begin{aligned} e^{Bt_i}X(t_i) - e^{Bt_{i-1}}X(t_{i-1}) &= \int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) \\ e^{Bt_i}X(t_i) &= e^{Bt_{i-1}}X(t_{i-1}) + \int_{t_{i-1}}^{t_i} e^{Bs}\gamma dW(s) \\ \text{Var}\left(e^{Bt_i}X(t_i)\right) &= \left(e^{Bt_{i-1}}\right)^2 \text{Var}(X(t_{i-1})) + \text{Var}\left(\int_0^t e^{Bs}\gamma dW(s)\right) \\ e^{2Bt_t}\text{Var}\left(X(t_i)\right) &= 0 + \frac{\gamma^2}{2B}\left[e^{2Bt_i} - e^{2Bt_{i-1}}\right] \\ \text{Var}\left(X(t_t)\right) &= \frac{\gamma^2}{2B}\left[1 - e^{-2B(t_i - t_{i-1})}\right] \\ \text{Var}\left(X(t_t)\right) &= \frac{\gamma^2}{2B}\left[1 - e^{-2B(\Delta t)}\right] \end{aligned}$$

As before, the values from the continuous system can be related to the discrete system. Since $\text{Var}(x_i) = \sigma_\eta^2$ and $X(t_i) = x_i$, then $\text{Var}(x_i) = \text{Var}(X(t_i)) = \frac{\gamma^2}{2B}\left[1 - e^{-2B(\Delta t)}\right] = \sigma_\eta^2$.

B.2 The EM method for DLMs

B.2.1 Derivation of the E-step

For each iteration of the EM algorithm above, $\mathbb{E}_{x_i|\boldsymbol{y}_0^t,\boldsymbol{\theta}_j}[x_i]$, $\mathbb{E}_{x_i|\boldsymbol{y}_0^t,\boldsymbol{\theta}_j}[x_i^2]$, and $\mathbb{E}_{x_i|\boldsymbol{y}_0^t,\boldsymbol{\theta}_j}[x_i x_{i-1}]$ must be computed. Unlike section 2.3.3, to provide more clarity to the derivations, the notations $(\hat{x}_i)_j$, $\left(\widehat{x_i^2}\right)_j$, and $\left(\widehat{x_{i-1}x_i}\right)_j$ were not used. Since each x_i will be condition on $\boldsymbol{\theta}_j$, the index j was dropped from $\boldsymbol{\theta}$. To compute the first of the required expectations, three steps considering partial data were be used. The partial data $(y_i, y_{i+1}, ..., y_{j-1}, y_j)$ where $j > i$ was be denoted as $\boldsymbol{y}_i^j$. The first step was to the compute the distribution of the state forecast which considers all past data. Mathematically, this was $x_i|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta}$ and called a_i. The second step was to include the current data and is called the state update or $x_i|\boldsymbol{y}_1^i, \boldsymbol{\theta}$. This was represented by the random variable α_i. Finally, all future data was included. This was $x_i|\boldsymbol{y}_1^T, \boldsymbol{\theta}$ and was be represented by β_i.

B.2.1.1 Initialization of the state forecast

Since there is no prior data for $t = 0$, computing $a_1 = x_1|\boldsymbol{y}_1^t$ is slightly different. Therefore, a prior must be placed on x_1 to take the place of the state forecast. A normal prior was placed on $p(x_1)$ as it is the conjugate distribution in this case. To maintain generality, let $p(x_0) \propto N(\mu_0, \sigma_0^2)$. The expectation ($\mathbb{E}(a_1)$) and the variance ($\text{Var}(a_1)$) follow immediately.

$$\mathbb{E}(a_1) = \mu_0 \tag{B.2}$$

$$\text{Var}(a_1) = \sigma_\eta^2 \tag{B.3}$$

B.2.1.2 The state forecast

From above, the state forecast (a_t) was defined as $a_t = x_t|\boldsymbol{y}_1^{t-1}$. Since the distribution is the result of iterative normals, a_t also has a normal distribution. Therefore, it was sufficient to compute the mean and variance. To find these, the following identities were needed which are just cases of the law of total expectations and law of total variance respectively. Consider $\mathbb{E}_{x_{t-1}|\boldsymbol{y}_1^{t-1}}\left\{\mathbb{E}_{x_t|x_{t-1},\boldsymbol{y}_1^{t-1}}[x_t|x_{t-1},\boldsymbol{y}_1^{t-1}]\right\}$ which was denoted as $\mathbb{E}_{x_{t-1}|\boldsymbol{y}_1^{t-1}}\left\{\mathbb{E}_{x_t|x_{t-1},\boldsymbol{y}_1^{t-1}}[x_t]\right\}$.

$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\int_{-\infty}^{\infty} x_i p(x_i|x_{i-1},\boldsymbol{y}_1^{i-1})dx_i\right\}$$
$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x_i p(x_i|x_{i-1},\boldsymbol{y}_1^{i-1})dx_i\right\} p(x_{i-1}|\boldsymbol{y}_1^{i-1})dx_{i-1}$$
$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x_i p(x_i|x_{i-1},\boldsymbol{y}_1^{i-1})p(x_{i-1}|\boldsymbol{y}_1^{i-1})dx_i\right\} dx_{i-1}$$
$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x_i p(x_i,x_{i-1}|\boldsymbol{y}_1^{i-1})dx_i\right\} dx_{i-1}$$

Since all functions above are continuous and integrable over the domain, then the order of integration can be switched.

$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \int\limits_{-\infty}^{\infty}\left\{\int\limits_{-\infty}^{\infty} x_i p(x_i,x_{i-1}|\boldsymbol{y}_1^{i-1})dx_{i-1}\right\} dx_i$$
$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \int_{-\infty}^{\infty}\left\{x_i p(x_i|\boldsymbol{y}_1^{i-1})\right\} dx_i$$

$$\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} = \mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i] \tag{B.4}$$

To find the variance, the identity $\mathrm{Var}(x_i|\boldsymbol{y}_1^{i-1}) = \mathbb{E}_{x_i^2|\boldsymbol{y}_1^{i-1}}[x_i^2] - \left(\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i]\right)^2$ was used. Applying (B.4) to $\mathrm{Var}(x_i|\boldsymbol{y}_1^{i-1}) = \mathbb{E}_{x_i^2|\boldsymbol{y}_1^{i-1}}[x_i^2] - \left(\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i]\right)^2$ resulted in $\mathrm{Var}(x_i|\boldsymbol{y}_1^{i-1}) = \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i^2|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i^2]\right\} - \left(\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\}\right)^2$

$$\begin{aligned}
\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i^2|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i^2]\right\} - \left(\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\}\right)^2 \\
\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathrm{Var}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i|x_{i-1},\boldsymbol{y}_1^{i-1}] + \left(\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right)^2\right\} ... \\
&\quad ... - \left(\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{t-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\}\right)^2 \\
\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathrm{Var}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} ... \\
&\quad ... + \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\left(\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right)^2\right\} - \left(\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\}\right)^2 \\
\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathrm{Var}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} + \mathrm{Var}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left(\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right) \qquad \text{(B.5)}
\end{aligned}$$

Given $x_i|x_{i-1} \sim N(kx_{i-1}, \sigma_\eta^2)$, the mean of a_i was calculated using (B.4).

$$\begin{aligned}
\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i] &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i]\right\} \\
\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i] &= \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{kx_{i-1}\right\} \\
\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i] &= k\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}}\left\{x_{i-1}\right\} \\
\mathbb{E}_{x_i|\boldsymbol{y}_1^{i-1}}[x_i] &= k\mathbb{E}\left\{\alpha_{i-1}\right\} \\
\mathbb{E}[a_i] &= k\mathbb{E}\left\{\alpha_{i-1}\right\} \qquad \text{(B.6)}
\end{aligned}$$

(B.5) was applied to compute the variance.

$$
\begin{aligned}
&\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) = ...\\
&\mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}} \left\{ \mathrm{Var}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i] \right\} + \mathrm{Var}_{x_{i-1}|\boldsymbol{y}_1^{i-1}} \left(\mathbb{E}_{x_i|x_{i-1},\boldsymbol{y}_1^{i-1}}[x_i] \right)\\
&\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) = \mathbb{E}_{x_{i-1}|\boldsymbol{y}_1^{i-1}} \left\{ \sigma_\eta^2 \right\} + \mathrm{Var}_{x_{i-1}|\boldsymbol{y}_1^{i-1}} \left(kx_{i-1} \right)\\
&\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) = \sigma_\eta^2 + k^2 \mathrm{Var}_{x_{i-1}|\boldsymbol{y}_1^{i-1}} \left(x_{i-1} \right)\\
&\mathrm{Var}_{x_i|\boldsymbol{y}_1^{i-1}}(x_i) = \sigma_\eta^2 + k^2 \mathrm{Var} \left(\alpha_{i-1} \right)\\
&\mathrm{Var}(a_i) = \sigma_\eta^2 + k^2 \mathrm{Var} \left(\alpha_{i-1} \right) \qquad \text{(B.7)}
\end{aligned}
$$

The for $i = 2$ to T, the expectation and variance were calculated using (B.6) and (B.7) respectively.

B.2.1.3 Calculate the state update

The state update was defined as $\alpha_i = x_i|\boldsymbol{y}_1^i$.

$$
\begin{aligned}
\alpha_i &= p(x_i|\boldsymbol{y_1^i})\\
\alpha_i &= p(x_i|y_i\boldsymbol{y_1^{i-1}})\\
\alpha_i &= \frac{p(y_i|x_i,\boldsymbol{y_1^i})p(x_i|\boldsymbol{y_1^{i-1}})}{p(y_i|\boldsymbol{y_1^i})}\\
\alpha_i &\propto p(y_i|x_i,\boldsymbol{y_1^i})p(x_i|\boldsymbol{y_1^{i-1}})
\end{aligned}
$$

$y_i|x_i, \boldsymbol{y}_1^i$ is conditionally independent of $\boldsymbol{y}_1^i$ so $p(y_i|x_i, \boldsymbol{y_1^i}) = p(y_i|x_i)$.

$$
\begin{aligned}
&\alpha_i \propto p(y_i|x_i)p(x_i|\boldsymbol{y_1^{i-1}})\\
&\alpha_i \propto \frac{1}{\sqrt{2\pi\sigma_\varsigma^2}} \exp\left[-\frac{1}{2\sigma_\varsigma^2} \left(y_i - x_i \right)^2 \right] \frac{1}{\sqrt{2\pi \mathrm{Var}(a_i)}} \exp\left[-\frac{1}{2\mathrm{Var}(a_t)} \left(x_i - \mathbb{E}(a_i) \right)^2 \right]
\end{aligned}
$$

$$
\begin{aligned}
\alpha_i &\propto \exp\left[-\frac{1}{2\sigma_\varsigma^2\text{Var}(a_i)}\Big(x_i^2(\sigma_\varsigma^2+\text{Var}(a_i)) - 2x_i(\sigma_\varsigma^2(a_i)+\text{Var}(a_i)y_i)\Big)\right] \times ... \\
&... \times \exp\left[-\frac{y_i^2}{2\sigma_\varsigma^2} - \frac{\mathbb{E}(a_i)}{2\text{Var}(a_i)}\right] \\
\alpha_i &\propto \exp\left[-\frac{\sigma_\varsigma^2+\text{Var}(a_i)}{2\sigma_\varsigma^2\text{Var}(a_i)}\left(x_i - \frac{\sigma_\varsigma^2\mathbb{E}(a_i)+\text{Var}(a_i)y_i}{\sigma_\varsigma^2+\text{Var}(a_i)}\right)^2\right]
\end{aligned}
$$

Which is the kernel of the normal distribution. The mean was reorganized in terms of its own variance.

$$
\alpha_i \propto N\left(\frac{\sigma_\varsigma^2\mathbb{E}(a_i)+\text{Var}(a_i)y_i}{\sigma_\varsigma^2+\text{Var}(a_i)}, \frac{\sigma_\varsigma^2\text{Var}(a_i)}{\sigma_\varsigma^2+\text{Var}(a_i)}\right) \tag{B.8}
$$

$$
\mathbb{E}(\alpha_i) = \frac{\sigma_\varsigma^2\mathbb{E}(a_i)+\text{Var}(a_i)y_i}{\sigma_\varsigma^2+\text{Var}(a_i)}
$$

$$
\mathbb{E}(\alpha_i) = \mathbb{E}(a_i) + \frac{\text{Var}(\alpha_i)}{\sigma_\varsigma^2}(y_i - \mathbb{E}(a_i)) \tag{B.9}
$$

$$
\text{Var}(\alpha_i) = \frac{\sigma_\varsigma^2\text{Var}(a_i)}{\sigma_\varsigma^2+\text{Var}(a_i)} \tag{B.10}
$$

For each $i < T$, return to steps outlined in appendix B.2.1.2.

B.2.1.4 Including all data

After a_i and α_i were computed for 1 to T, one can compute $\beta_i = x_i|\boldsymbol{y}_1^T$. This was done in reverse order, starting with $i = T$. The case where $i = T$ required no additional calculation because $\alpha_T = x_T|\boldsymbol{Y}_1^T = \beta_T$. For all other i, additional computation was needed to include $\boldsymbol{y}_{i+1}^T$.

$$
\begin{aligned}
p(x_i|x_{i+1},\boldsymbol{y}_1^T) &= p(x_i|x_{i+1},\boldsymbol{y}_1^i) \\
p(x_i|x_{i+1},\boldsymbol{y}_1^T) &\propto p(x_{i+1}|x_i,\boldsymbol{y}_1^i)p(x_i|\boldsymbol{y}_1^i) \\
p(x_i|x_{i+1},\boldsymbol{y}_1^T) &\propto p(x_{i+1}|x_i)p(x_i|\boldsymbol{y}_1^i)
\end{aligned}
$$

$$p(x_i|x_{i+1}, \boldsymbol{y}_1^T) \propto \exp\left\{-\frac{1}{2\sigma_\eta^2}(x_{i+1} - kx_t)^2 - \frac{1}{2\text{Var}(\alpha_i)}(x_i - \mathbb{E}(\alpha_i))^2\right\}$$
$$p(x_i|x_{i+1}, \boldsymbol{y}_1^T) \propto N_{x_i}\left(\frac{\text{Var}(\alpha_t)kx_{i+1} + \sigma_\eta^2\mathbb{E}(\alpha_i)}{\text{Var}(\alpha_i)k^2 + \sigma_\eta^2}, \frac{\sigma_\eta^2\text{Var}(\alpha_i)}{\text{Var}(\alpha_i)k^2 + \sigma_\eta^2}\right) \tag{B.11}$$

$\frac{\text{Var}(\alpha_i)kx_{i+1}+\sigma_\eta^2\mathbb{E}(\alpha_i)}{\text{Var}(\alpha_i)k^2+\sigma_\eta^2}$ was re-written as $\mathbb{E}(\alpha_i) + \frac{\text{Var}(\alpha_i)k}{\text{Var}(a_{i+1})}[x_{i+1} - \mathbb{E}(a_{i+1})]$ and $\frac{\sigma_\eta^2\text{Var}(\alpha_i)}{\text{Var}(\alpha_i)k^2+\sigma_\eta^2}$ was re-written as $\frac{\sigma_\eta^2\text{Var}(\alpha_i)}{\text{Var}(a_{i+1})}$.

$$p(x_i|x_{i+1}, \boldsymbol{y}_1^T) \propto N_{x_i}\left(\mathbb{E}(\alpha_i) + \frac{\text{Var}(\alpha_i)k}{\text{Var}(a_{i+1})}[x_{i+1} - \mathbb{E}(a_{i+1})], \frac{\sigma_\eta^2\text{Var}(\alpha_i)}{\text{Var}(a_{i+1})}\right) \tag{B.12}$$

Using (B.4), $\mathbb{E}(\beta_i)$. was calculated.

$$\begin{aligned}
\mathbb{E}(\beta_i) &= \mathbb{E}_{x_i|\boldsymbol{y}_1^T}\left[x_i\right] \\
\mathbb{E}(\beta_i) &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{\mathbb{E}_{x_t|x_{t+1},\boldsymbol{y}_1^T}\left[x_t\right]\right\} \\
\mathbb{E}(B_i) &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{\mathbb{E}(\alpha_i) + \frac{\text{Var}(\alpha_i)k}{\text{Var}(a_{i+1})}[x_{i+1} - \mathbb{E}(a_{i+1})]\right\} \\
\mathbb{E}(B_i) &= \mathbb{E}(\alpha_i) + \frac{\text{Var}(\alpha_i)k}{\text{Var}(a_{i+1})}[\mathbb{E}(\beta_{i+1}) - \mathbb{E}(a_{i+1})] \qquad \text{(B.13)}
\end{aligned}$$

B.2.1.5 $\mathbb{E}(x_i^2|\boldsymbol{y}_1^T)$ and $\mathbb{E}(x_i x_{i-1}|\boldsymbol{y}_1^T)$

Recall $x_i^2|\boldsymbol{y}_1^T = \beta_i^2$. Since $\text{Var}(\beta_i) = \mathbb{E}[\beta_i^2] - (\mathbb{E}[\beta_i])^2$. Therefore computing $\text{Var}(\beta_i)$ was sufficient to find $\mathbb{E}[\beta_i^2]$. To find $\text{Var}(\beta_i)$, (B.5) was applied.

$$\begin{aligned}
\text{Var}(\beta_i) &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{\text{Var}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} + \text{Var}_{x_{i+1}|\boldsymbol{y}_1^T}\left(\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^i}[x_i]\right) \\
\text{Var}(\beta_i) &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{\frac{\sigma_\eta^2\text{Var}(\alpha_i)}{\text{Var}(a_{i+1})}\right\}...
\end{aligned}$$

$$
\begin{aligned}
&\quad ... + \mathrm{Var}_{x_{i+1}|\boldsymbol{y}_1^T}\left(\mathbb{E}(\alpha_i) + \frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}[x_{i+1} - \mathbb{E}(a_{i+1})]\right) \\
\mathrm{Var}(\beta_i) &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{\frac{\sigma_\eta^2\mathrm{Var}(\alpha_i)}{\mathrm{Var}(a_{i+1})}\right\} + \mathrm{Var}_{x_{i+1}|\boldsymbol{y}_1^T}\left(\frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}x_{i+1}\right) \\
\mathrm{Var}(\beta_i) &= \frac{\sigma_\eta^2\mathrm{Var}(\alpha_t)}{\mathrm{Var}(a_{i+1})} + \left(\frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}\right)^2 \mathrm{Var}_{x_{i+1}|\boldsymbol{y}_1^T}[x_{i+1}] \\
\mathrm{Var}(\beta_i) &= \frac{\sigma_\eta^2\mathrm{Var}(\alpha_i)}{\mathrm{Var}(a_{i+1})} + \left(\frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}\right)^2 \mathrm{Var}(\beta_{i+1}) \\
\mathrm{Var}(\beta_i) &= \frac{(\mathrm{Var}(a_{i+1}))\,\mathrm{Var}(\alpha_i)}{\mathrm{Var}(a_{i+1})} + ... \\
&\quad ... - \frac{k^2\mathrm{Var}(\alpha_i)^2}{\mathrm{Var}(a_{i+1})^2}\left(\mathrm{Var}(a_{i+1})\right) + \left(\frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}\right)^2 \mathrm{Var}(\beta_{i+1}) \\
\mathrm{Var}(\beta_i) &= \mathrm{Var}(\alpha_i) + \left(\frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}\right)^2 [\mathrm{Var}(\beta_{i+1}) - \mathrm{Var}(a_{i+1})] \qquad \text{(B.14)}
\end{aligned}
$$

With (B.14), $\mathbb{E}[\beta_i^2]$ was found.

$$
\mathbb{E}[\beta_i^2] = \mathrm{Var}(\beta_i) + \mathbb{E}[\beta_i]^2 \qquad \text{(B.15)}
$$

To compute $\mathbb{E}[x_{i+1}x_i|\boldsymbol{y}_1^T]$, $\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\}$ was considered.

$$
\begin{aligned}
\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\int_{-\infty}^{\infty} x_i p(x_i|x_{i+1},\boldsymbol{y}_1^T)dx_i\right\} \\
\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} &= \int_{-\infty}^{\infty}\left\{x_{i+1}\int_{-\infty}^{\infty} x_i p(x_i|x_{i+1},\boldsymbol{y}_1^T)dx_i\right\} \times p(x_{i+1}|\boldsymbol{y}_1^T)dx_{i+1} \\
\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x_{i+1}x_i p(x_i|x_{i+1},\boldsymbol{y}_1^T)p(x_{i+1}|\boldsymbol{y}_1^T)dx_i\right\}dx_{i+1} \\
\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} x_{i+1}x_i p(x_i,x_{t+1}|\boldsymbol{y}_1^T)dx_t\right\}dx_{i+1} \\
\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}[x_i]\right\} &= \mathbb{E}_{x_{i+1},x_i|\boldsymbol{y}_1^T}[x_{i+1}x_i]
\end{aligned}
$$

Then $\mathbb{E}[\beta_{i+1}\beta_t]$ was solved for using $\mathbb{E}[x_{i+1}x_i|\boldsymbol{y}_1^T] = \mathbb{E}[\beta_{i+1}\beta_t]$.

$$\mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}\left[x_i\right]\right\} = \mathbb{E}[\beta_{i+1}\beta_i] \tag{B.16}$$

Then, (B.16) was used to find $\mathbb{E}[\beta_{i+1}\beta_i]$.

$$\begin{aligned}
\mathbb{E}[\beta_{i+1}\beta_i] &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}_{x_i|x_{i+1},\boldsymbol{y}_1^T}\left[x_i\right]\right\} \\
\mathbb{E}[\beta_{i+1}\beta_i] &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\left(\mathbb{E}(\alpha_i) + \frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}[x_{i+1} - \mathbb{E}(a_{i+1})]\right)\right\} \\
\mathbb{E}[\beta_{i+1}\beta_i] &= \mathbb{E}_{x_{i+1}|\boldsymbol{y}_1^T}\left\{x_{i+1}\mathbb{E}(\alpha_i) + \frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}[x_{i+1}^2 - \mathbb{E}(a_{i+1})x_{i+1}]\right\} \\
\mathbb{E}[\beta_{i+1}\beta_i] &= \mathbb{E}(\beta_{i+1})\mathbb{E}(\alpha_i) + \frac{\mathrm{Var}(\alpha_i)k}{\mathrm{Var}(a_{i+1})}[\mathbb{E}(\beta_{i+1}^2) - \mathbb{E}(a_{i+1})\mathbb{E}(\beta_{i+1})]
\end{aligned} \tag{B.17}$$

B.2.2 Derivation of the M-step

For j^{th} the M-step, the $\hat{\boldsymbol{\theta}}j$ that maximized the $\mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[p(\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\boldsymbol{\theta})]$ was found. Therefore, the maximum likelihood estimator (MLE) was computed for each element of $\boldsymbol{\theta}$. For this $\ln p(\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\boldsymbol{\theta})$ was revisited below.

$$\begin{aligned}
\ln(p(\boldsymbol{x}_1^T,\boldsymbol{y}_1^T|\boldsymbol{\theta})) &= \ln\left\{p(x_1|\boldsymbol{\theta})\right\} + \sum_{i=2}^{T}\ln\left\{p(x_i|x_{i-1},\boldsymbol{\theta})\right\} + \sum_{i=1}^{T}\ln\left\{p(y_i|x_i,\boldsymbol{\theta})\right\} \\
\ln\left\{p(x_1|\boldsymbol{\theta})\right\} &= -\frac{1}{2}\ln(2\pi\sigma_\eta^2) - \frac{1}{2\sigma_\eta^2}(x_1 - \mu_0)^2 \\
\sum_{i=2}^{T}\ln\left\{p(x_i|x_{i-1},\boldsymbol{\theta})\right\} &= -\frac{T-1}{2}\ln(2\pi\sigma_\eta^2) - \sum_{i=2}^{T}\frac{1}{2\sigma_\eta^2}(x_i^2 - 2kx_ix_{i-1} + k^2x_{i-1}^2) \\
\sum_{i=1}^{T}\ln\left\{p(y_i|x_i,\boldsymbol{\theta})\right\} &= -\frac{T}{2}\ln(2\pi\sigma_\varsigma^2) - \sum_{i=1}^{T}\frac{1}{2\sigma_\varsigma^2}(y_i^2 - 2y_ix_i + x_i^2)\textbf{âĂć}
\end{aligned}$$

For brevity, the notations from section (2.3.3) were used.

B.2.2.1 MLE of μ_0

$$
\begin{aligned}
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \mu_0} &= -\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln\{p(x_1|\boldsymbol{\theta})\}]}{\partial \mu_0} \\
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \mu_0} &= -\frac{1}{2\sigma_\eta^2}\left(-2\left(\hat{x}_1\right)_j + 2\mu_0\right) \\
0 &= -\frac{1}{2\sigma_\eta^2}\left(-2\left(\hat{x}_1\right)_j + 2\left(\hat{\mu}_0\right)_j\right) \\
\left(\hat{\mu}_0\right)_j &= \left(\hat{x}_1\right)_j
\end{aligned}
\tag{B.18}
$$

From this, it is easy to see $\frac{\partial^2 \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \mu_0^2} = -\frac{1}{\sigma_\eta^2} < 0$. Therefore, the quantity found in (B.18) was indeed a maximum by the second derivative test.

B.2.2.2 MLE of k

$$
\begin{aligned}
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial k} &= -\frac{\partial\left(\sum_{i=2}^{T}\ln\left[\{p(x_i|x_{i-1},\boldsymbol{\theta})\}\right]\right)}{\partial k} \\
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial k} &= -\frac{1}{2\sigma_\eta^2}\left(-\sum_{i=2}^{T}2\left(\widehat{x_{i-1}x_i}\right)_j + 2\sum_{i=2}^{T}\left(\widehat{x_{i-1}^2}\right)_j\right) \\
0 &= -\frac{1}{2\sigma_\eta^2}\left(-\sum_{i=2}^{T}2\left(\widehat{x_{i-1}x_i}\right)_j + 2\sum_{i=2}^{T}\hat{k}_j\left(\widehat{x_{i-1}^2}\right)_j\right) \\
\hat{k}_j &= \frac{\sum_{i=2}^{T}\left(\widehat{x_{i-1}x_i}\right)_j}{\sum_{i=2}^{T}\left(\widehat{x_{i-1}^2}\right)_j}
\end{aligned}
\tag{B.19}
$$

Then, it can be confirmed that the critical point is a maximum through the second derivative test.

$$
\frac{\partial^2 \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial k^2} = -\frac{1}{2\sigma_\eta^2}2\sum_{i=2}^{T}\left(\widehat{x_{i-1}^2}\right)_j \tag{B.20}
$$

In (B.20), it can seen that $\frac{\partial^2 \mathbb{E}_{\Upsilon_{i-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial k^2} < 0$ since both σ_η^2 and $\left(\widehat{x_{i-1}^2}\right)_j$ must be positive, guaranteeing that the critical point is a maximum.

B.2.2.3 MLE of σ_η^2

$$
\begin{aligned}
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \sigma_\eta^2} &= -\frac{\partial\left(\mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}\left[\ln\left\{p(x_1|\boldsymbol{\theta})\right\} + \sum_{i=2}^{T} \ln\left\{p(x_i|x_{i-1},\boldsymbol{\theta})\right\}\right]\right)}{\partial \sigma_\eta^2} \\
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \sigma_\eta^2} &= -\frac{T-1}{2\sigma_\eta^2} + \frac{1}{2(\sigma_\eta^2)^2}\left[MSE\left((\hat{x}_i)_j, \hat{k}_j\,(\hat{x}_{i-1})_j\right)\right] ... \\
&\quad ... - \frac{1}{2\sigma_\eta^2} + \frac{1}{2(\sigma_\eta^2)^2}\left[MSE\left((\hat{x}_1)_j, (\mu_0)_j\right)\right] \\
(T)\left(\hat{\sigma}_\eta^2\right)_j &= MSE\left((\hat{x}_i)_j, \hat{k}_j\,(\hat{x}_{i-1})_j\right) + MSE\left((\hat{x}_1)_j, (\mu_0)_j\right) \\
\left(\hat{\sigma}_\eta^2\right)_j &= \frac{MSE\left((\hat{x}_i)_j, \hat{k}_j\,(\hat{x}_{i-1})_j\right) + MSE\left((\hat{x}_1)_j, (\hat{\mu}_0)_j\right)}{T} \qquad \text{(B.21)}
\end{aligned}
$$

It can be confirmed that the point is a maximum evaluating the second derivative at $\left(\hat{\sigma}_\eta^2\right)_j$.

B.2.2.4 MLE of σ_ς^2

$$
\begin{aligned}
\frac{\partial \mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}[\ln(p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}))]}{\partial \sigma_\varsigma^2} &= -\frac{\partial\left(\mathbb{E}_{\boldsymbol{x}_0^T|\boldsymbol{y}_0^T,\hat{\boldsymbol{\theta}}_{j-1}}\left[\sum_{i=1}^{T} \ln\left\{p(y_i|x_i,\boldsymbol{\theta})\right\}\right]\right)}{\partial \sigma_\varsigma^2} \\
0 &= -\frac{T}{2\left(\hat{\sigma}_\varsigma^2\right)_j} + \frac{1}{2\left(\hat{\sigma}_\varsigma^2\right)_j^2}\left[\sum_{i=1}^{T}\left(y_i^2 - 2y_i\,(\hat{x}_i)_j + \left(\widehat{x_i^2}\right)_j\right)\right] \\
\left(\hat{\sigma}_\varsigma^2\right)_j &= \frac{1}{T}\left[\sum_{i=1}^{T}\left(y_i^2 - 2y_i\,(\hat{x}_i)_j + \left(\widehat{x_i^2}\right)_j\right)\right] \qquad \text{(B.22)}
\end{aligned}
$$

Again, it can be confirmed that (B.22) was a maximum using the aforementioned methods.

B.2.3 Conditional posterior distribution for DLMs

B.2.3.1 Derivation of the conditional posterior of σ_η^2

$$\begin{aligned}
\sigma_\eta^2 &\sim \mathcal{IG}_{\sigma_\eta^2}\left(\frac{n_\eta}{2}, \frac{d_\eta}{2}\right) \\
p(\sigma_\eta^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\eta^2}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\sigma_\varsigma^2, \sigma_\eta^2, k, \mu_0)p(\sigma_\eta^2) \\
p(\sigma_\eta^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\eta^2}) &\propto p(x_1|\boldsymbol{\theta})\left[\prod_{i=2}^{T} p(x_i|x_{i-1}, \boldsymbol{\theta})\right] ... \\
&... \times \left[\prod_{i=1}^{T} p(y_i|x_i, \boldsymbol{\theta})\right] p(\sigma_\eta^2) \\
p(\sigma_\eta^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\eta^2}) &\propto (\sigma_\eta^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma_\eta^2}(x_i - \mu_0)^2\right\} ... \\
&... \times (\sigma_\eta^2)^{-\frac{T-1}{2}} \exp\left\{-\frac{1}{2\sigma_\eta^2}\left[\sum_{i=2}^{T}(x_i - kx_{i-1})^2\right]\right\} ... \\
&... \times (\sigma_\eta^2)^{-\frac{n_\eta}{2}-1} \exp\left\{-\frac{d_\eta}{2\sigma_\eta^2}\right\} \\
\sigma_\eta^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\eta^2} &\sim \mathcal{IG}\left(\frac{T + n_\eta}{2}, \frac{\left[d_\eta + (x_1 - \mu_0)^2 + \sum_{i=2}^{T}(x_i - kx_{i-1})^2\right]}{2}\right)
\end{aligned} \tag{B.23}$$

B.2.3.2 Derivation of the conditional posterior of σ_ς^2

$$\begin{aligned}
\sigma_\varsigma^2 &\sim \mathcal{IG}\left(\frac{n_\varsigma}{2}, \frac{d_\varsigma}{2}\right) \\
p(\sigma_\varsigma^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\varsigma^2}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\sigma_\varsigma^2, \sigma_\eta^2, k, \mu_0)p(\sigma_\varsigma^2) \\
p(\sigma_\varsigma^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\varsigma^2}) &\propto\propto p(x_1|\boldsymbol{\theta})\left[\prod_{i=2}^{T} p(x_i|x_{i-1}, \boldsymbol{\theta})\right]... \\
&... \times \left[\prod_{i=1}^{T} p(y_i|x_i, \boldsymbol{\theta})\right] p(\sigma_\varsigma^2) \\
p(\sigma_\varsigma^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\varsigma^2}) &\propto (\sigma_\varsigma^2)^{-\frac{T}{2}} \exp\left\{-\frac{1}{2\sigma_\varsigma^2}\left[\sum_{i=1}^{T}(y_i - x_i)^2\right]\right\}... \\
&... \times (\sigma_\varsigma^2)^{-\frac{n_\varsigma}{2}-1} \exp\left\{-\frac{d_\varsigma}{2\sigma_\eta^2}\right\} \\
\sigma_\varsigma^2|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\sigma_\varsigma^2} &\sim \mathcal{IG}\left(\frac{T + n_\varsigma}{2}, \frac{1}{2}\left[d_\varsigma + \sum_{i=1}^{T}(y_i - x_i)^2\right]\right) \qquad \text{(B.24)}
\end{aligned}$$

B.2.3.3 Derivation of the conditional posterior of k

$$\begin{aligned}
p(k) &= I[0 \leq k \leq 1] \\
p(k|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-k}) &\propto p(\boldsymbol{y}_1^T, \boldsymbol{x}_1^T|k, \mu_0, \sigma_\eta^2, \sigma_\varsigma^2)p(k) \\
p(k|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-k}) &\propto \left[\prod_{i=2}^{T} p(x_i|x_{i-1}, \boldsymbol{\theta})\right] \times I[-1 \leq k \leq 1] \\
p(k|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-k}) &\propto \exp\left\{-\frac{1}{2\sigma_\eta^2}\left[\sum_{i=2}^{T}\left(x_i - kx_{i-1}\right)^2\right]\right\} I[-1 \leq k \leq 1] \\
p(k|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-k}) &\propto \exp\left\{-\frac{1}{2\sigma_\eta^2}\left[k^2\sum_{i=2}^{T} x_{i-1}^2 - k\sum_{i=2}^{T} x_i x_{i-1}\right]\right\} \times I[-1 \leq k \leq 1] \\
p(k|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-k}) &\propto \exp\left\{-\frac{\sum_{i=2}^{T} x_{i-1}^2}{2\sigma_\eta^2}\left(k - \frac{\sum_{i=2}^{T} x_i x_{i-1}}{\sum_{i=2}^{T} x_{i-1}}\right)^2\right\} \times I[-1 \leq k \leq 1] \qquad \text{(B.25)}
\end{aligned}$$

B.2.3.4 Derivation of the conditional posterior of μ_0

$$
\begin{aligned}
\mu_0 &\sim \mathcal{N}\left(m_0, \sigma_m^2\right) \\
p(\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0}) &\propto p(\boldsymbol{x}_1^T, \boldsymbol{y}_1^T|\sigma_\varsigma^2, \sigma_\eta^2, k, \mu_0)p(\mu_0) \\
p(\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0}) &\propto p(x_1|\boldsymbol{\theta})p(\mu_0) \\
p(\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0}) &\propto \exp\left\{-\frac{1}{2\sigma_\eta^2}\left(x_1 - \mu_0\right)^2 - \frac{1}{2\sigma_m^2}\left(\mu_0 - m_0\right)^2\right\} \\
p(\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0}) &\propto \exp\left\{\frac{-1}{2\sigma_\eta^2\sigma_m^2}\left[\mu_0^2\left(\sigma_m^2 + \sigma_\eta^2\right) - 2\mu_0\left(x_1\sigma_m^2 + m_0\sigma_\eta^2\right)\right]\right\} \\
p(\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0}) &\propto \exp\left\{-\frac{\sigma_m^2 + \sigma_\eta^2}{2\sigma_\eta^2\sigma_m^2}\left(\mu_0 - \frac{x_1\sigma_m^2 + m_0\sigma_\eta^2}{\sigma_m^2 + \sigma_\eta^2}\right)^2\right\} \\
\mu_0|\boldsymbol{x}_1^T, \boldsymbol{y}_1^T \boldsymbol{\theta}_{-\mu_0} &\sim \mathcal{N}\left(\frac{x_1\sigma_m^2 + m_0\sigma_\eta^2}{\sigma_m^2 + \sigma_\eta^2}, \frac{\sigma_\eta^2\sigma_m^2}{\sigma_m^2 + \sigma_\eta^2}\right) \qquad \text{(B.26)}
\end{aligned}
$$

B.2.3.5 Expansion of $\epsilon_{B,j}$

$$
f(k) = -\frac{\ln k}{\Delta t} \qquad \text{(B.27)}
$$

Then behavior of $\epsilon_{B,j}$ was studied using an an expansion around the true value of k where $f(k)$ was used to denote the right hand side of (B.27).

$$
\begin{aligned}
B_j &= -\frac{\ln(k_j)}{\Delta t} \\
B_j &= -\frac{\ln(k + \epsilon_{k,j})}{\Delta t} \\
B_j &= f(k) - \left(\frac{1}{k\Delta t}\right)(\epsilon_{k,j}) + \mathcal{O}(\epsilon_{k,j}^2)
\end{aligned}
$$

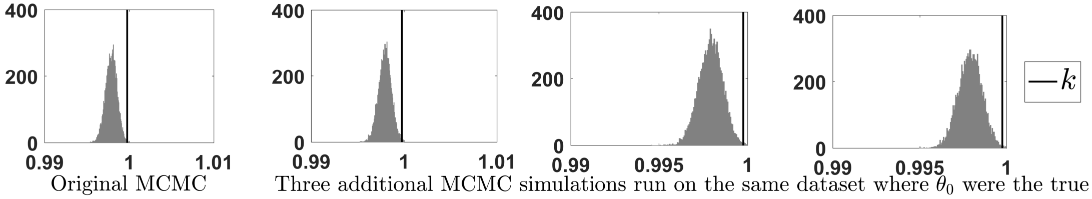
This was then used to approximate $\epsilon_{B,j}$ for each $\epsilon_{k,j}$.

$$
\begin{aligned}
B_j &= f(k) - \left(\frac{1}{k\Delta t}\right)(\epsilon_{k,j}) + \mathcal{O}(\epsilon_{k,j}^2) \\
B + \epsilon_{B,j} &= f(k) - \left(\frac{1}{k\Delta t}\right)(\epsilon_{k,j}) + \mathcal{O}(\epsilon_{k,j}^2) \\
f(k) + \epsilon_{B,j} &= f(k) - \left(\frac{1}{k\Delta t}\right)(\epsilon_{k,j}) + \mathcal{O}(\epsilon_{k,j}^2) \\
\epsilon_{B,j} &= -\left(\frac{1}{k\Delta t}\right)(\epsilon_{k,j}) + \mathcal{O}(\epsilon_{k,j}^2) \qquad \text{(B.28)}
\end{aligned}
$$

B.2.3.6 Additionl MCMC simulations for the dataset from section 2.3.5 for $\Delta t = \frac{1}{4096}$

Three additional MCMC simulations for $\Delta t = \frac{1}{4096}$ were run on the dataset where $\Delta t = \frac{1}{4096}$ from section 2.3.5. The true values were used for $\boldsymbol{\theta}_0$ for each addition simulation. The histograms for $\{k_j\}$, $\left\{\left(\sigma_\eta^2\right)_j\right\}$, and $\left(\sigma_\varsigma^2\right)_j$ were plotted in figures B.1, B.2, and B.3 respectively.

Histograms of the posterior draws from $\{k_j\}$ against the true value of k



Original MCMC

Three additional MCMC simulations run on the same dataset where θ_0 were the true θ

Figure B.1: Histograms of $\{k_j\}$ for the dataset from section 2.3.5 for $\Delta t = \frac{1}{4096}$

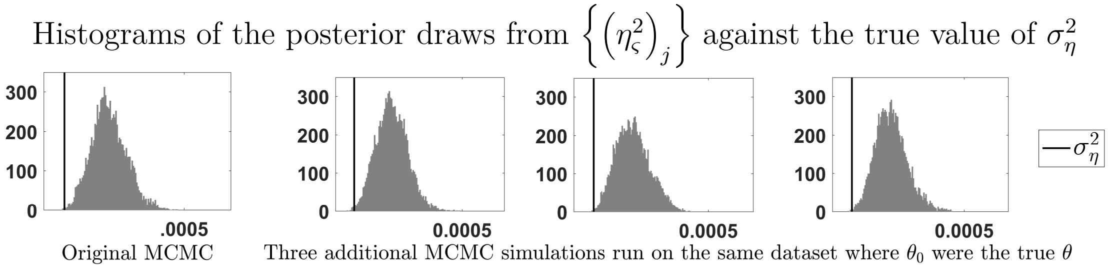


Figure B.2: Histograms of $\left\{\left(\sigma_{\eta}^2\right)_j\right\}$ for the dataset from section 2.3.5 for $\Delta t = \frac{1}{4096}$

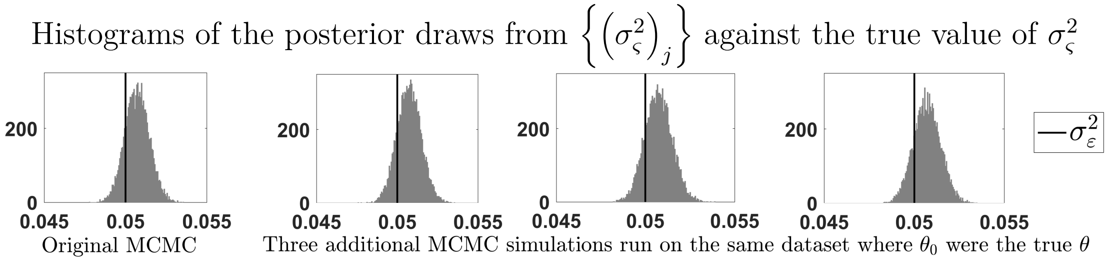


Figure B.3: Histograms of $\left\{\left(\sigma_{\varsigma}^2\right)_j\right\}$ from four independent MCMC simulations for the dataset from section 2.3.5 for $\Delta t = \frac{1}{4096}$

Appendix C

Supplement: Inference on a discrete state space CTMC

C.1 Derivation: Mapping a 2 state CTMC to a 2 state Markov Model

(C.1) System of differential equations representation of the discrete two state CTMC

$$\frac{dP(t)_1}{dt} = -r_{12}P(t)_1 + r_{21}P(t)_2 \tag{C.2}$$

$$\frac{dP(t)_2}{dt} = r_{12}P(t)_1 - r_{21}P(t)_2 \tag{C.3}$$

To represent (C.1) as a discrete system, the differential equations must be solved. To illustrate this (C.2) was used to solve for q_{12} or $P(X_{i+1} = 2|X_i = 1)$.

$$
\begin{aligned}
\frac{dP(t)_1}{dt} &= -r_{12}P(t)_1 + r_{21}P(t)_2 \\
\frac{dP(t)_1}{dt} &= -r_{12}P(t)_1 + r_{21}(1 - P(t)_1) \\
\frac{dP(t)_1}{dt} + (r_{12} + r_{21})P(t)_1 &= r_{21} \\
\int_{t_i}^{t_{i+1}} \left[P(t)_1 e^{(r_{12}+r_{21})t}\right]_t dt &= \int_{t_i}^{t_{i+1}} r_{21}e^{(r_{12}+r_{21})t}dt
\end{aligned}
$$

To understand the case where $X(t_i) = 1$, the equivalent statement $P(t_i)_1 = 1$ was substituted.

$$
\begin{aligned}
P(t_{i+1})_1 e^{(r_{12}+r_{21})t_{i+1}} - \left(e^{(r_{12}+r_{21})t_i}\right) &= \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})t_{i+1}} - \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})t_i} \\
P(t_{i+1})_1 e^{(r_{12}+r_{21})(t_{i+1}-t_i)} - 1 &= \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})(t_{i+1}-t_i)} - \frac{r_{21}}{r_{12}+r_{21}}
\end{aligned}
$$

For brevity, let $t_{i+1} - t_i = \Delta t$.

$$
\begin{aligned}
P(t_{i+1})_1 e^{(r_{12}+r_{21})\Delta t} &= \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})\Delta t} - \frac{r_{21}}{r_{12}+r_{21}} + 1 \\
P(t_{i+1})_1 e^{(r_{12}+r_{21})\Delta t} &= \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})\Delta t} + \frac{r_{12}}{r_{12}+r_{21}} \\
P(t_{i+1})_1 e^{(r_{12}+r_{21})\Delta t} &= \frac{r_{21}}{r_{12}+r_{21}}e^{(r_{12}+r_{21})\Delta t} + \frac{r_{12}}{r_{12}+r_{21}} \\
P(t_{i+1})_1 &= \frac{r_{21}}{r_{12}+r_{21}} + \frac{r_{12}}{r_{12}+r_{21}}e^{-(r_{12}+r_{21})\Delta t} \qquad \text{(C.4)}
\end{aligned}
$$

Since the assumption $X_i = 1$ was made, than $P(t_{i+1})_1 = P(X_{i+1} = 1|X_i = 1)$. Furthermore, since the system can only stay in S_1 or change to S_{2A}, $P(t_{i+1})_1 = 1 - P(X_{i+1} = 2|X_i = 1)$.

$$
\begin{aligned}
1 - P(X_{i+1} = 2|X_i = 1) &= \frac{r_{21}}{r_{12} + r_{21}} + \frac{r_{12}}{r_{12} + r_{21}} e^{-(r_{12}+r_{21})\Delta t} \\
P(X_{i+1} = 2|X_i = 1) &= 1 - \frac{r_{21}}{r_{12} + r_{21}} - \frac{r_{12}}{r_{12} + r_{21}} e^{-(r_{12}+r_{21})\Delta t} \\
P(X_{i+1} = 2|X_i = 1) &= \frac{r_{12}}{r_{12} + r_{21}} - \frac{r_{12}}{r_{12} + r_{21}} e^{-(r_{12}+r_{21})\Delta t} \\
P(X_{i+1} = 2|X_i = 1) &= \frac{r_{12}}{r_{12} + r_{21}} \left(1 - e^{-(r_{12}+r_{21})\Delta t}\right) \qquad \text{(C.5)}
\end{aligned}
$$

Following the same process, one can derive $P(X_{i+1} = 1|X_i = 2)$.

$$
P(X_{i+1} = 1|X_i = 2) = \frac{r_{21}}{r_{12} + r_{21}} \left(1 - \exp(-\Delta t(r_{12} + r_{21}))\right) \qquad \text{(C.6)}
$$

Finally, r_{12} and r_{21} can be found given $P(X_{i+1} = k|X_i = j)$ from (C.5) and (C.6).

$$
\begin{aligned}
r_{12} + r_{21} &= -\frac{\ln(1 - (P(X_{i+1} = 2|X_i = 1) + P(X_{i+1} = 1|X_i = 2)))}{\Delta t} && \text{(C.7)} \\
r_{12} &= \frac{P(X_{i+1} = 2|X_i = 1)(r_{12} + r_{21})}{1 - \exp(-\Delta t(r_{12} + r_{21}))} && \text{(C.8)} \\
r_{21} &= \frac{P(X_{i+1} = 1|X_i = 2)(r_{12} + r_{21})}{1 - \exp(-\Delta t(r_{12} + r_{21}))} && \text{(C.9)}
\end{aligned}
$$

C.2 Derivation: Mapping a 2 state Markov model to CMTC

Recall, $\boldsymbol{Q}(\Delta t)$ approximated given $\boldsymbol{R}$ using the first D terms of the matrix exponential or (C.10). Approximating $\boldsymbol{R}$ in terms of $\boldsymbol{Q}(\Delta t)$ required inversion of the matrix exponential. Inverting the matrix exponential given an approximation for $\boldsymbol{Q}(\Delta t)$ was not as straightforward as approximating $\boldsymbol{Q}$. To approximate, $\boldsymbol{R}$ was rewritten as in terms of $\boldsymbol{Q}$ and higher powers of $\boldsymbol{R}$ as listed in (C.11). $\boldsymbol{R}$ was approximated using an iterative process where $\boldsymbol{R}^{(D)}$ denotes an approximation with error $o(\Delta t^D)$

$$\boldsymbol{Q}(\Delta t) = \sum_{d=0}^{D} \frac{(\mathbf{R}(\Delta t))^d}{d!} + o(\Delta t^D) \tag{C.10}$$

$$\boldsymbol{R} = \frac{\boldsymbol{Q} - \boldsymbol{I}}{\Delta t} - \frac{\boldsymbol{R}^2 \Delta t}{2!} - \frac{\boldsymbol{R}^3 \Delta t^2}{3!} - \frac{\boldsymbol{R}^4 \Delta t^3}{4!} - \dots \tag{C.11}$$

The first was an approximation with error $o(1)$ or $o(\Delta t^0)$. For this, $\boldsymbol{R}^{(0)}$ was set equal to all terms of order less than or equal to Δt^0 as shown in (C.12).

$$\boldsymbol{R}^{(0)} = \frac{\boldsymbol{Q} - \boldsymbol{I}}{\Delta t} \tag{C.12}$$

For the second iteration, $\boldsymbol{R}^{(0)}$ was substituted for $\boldsymbol{R}$ on the right side of (C.11). Then $\boldsymbol{R}^{(1)}$ was found by including all terms with degree less than or equal to Δt^1 from (C.11) and was listed in (C.13).

$$\begin{aligned} \boldsymbol{R}^{(1)} &= \frac{\boldsymbol{Q} - \boldsymbol{I}}{\Delta t} - \frac{\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t}{2!} \\ \boldsymbol{R}^{(1)} &= \boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t}{2!} \end{aligned} \tag{C.13}$$

It can be verified that the error of (C.13) is $o(\Delta t^1)$ by substituting $\boldsymbol{R}^{(1)}$ into (C.10).

For the third iteration, $\boldsymbol{R}^{(1)}$ was substituted for $\boldsymbol{R}$ on the right side of (C.11). Then $\boldsymbol{R}^{(2)}$ was found by including all terms with degree less than or equal to Δt^2 from (C.11) and was listed as (C.14). The error of $o(\Delta t^2)$ could be confirmed as before by substituting $\boldsymbol{R}^{(2)}$ from (C.14) into $\boldsymbol{R}$ in (C.10). .

$$\begin{aligned}
\boldsymbol{R}^{(2)} &= \frac{\boldsymbol{Q}-\boldsymbol{I}}{\Delta t} - \frac{\left(\boldsymbol{R}^{(1)}\right)^2 \Delta t}{2!} - \frac{\left(\boldsymbol{R}^{(1)}\right)^3 \Delta t^2}{3!} \\
\boldsymbol{R}^{(2)} &= \boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t}{2!}\right)^2 \Delta t}{2!} - \frac{\left(\boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t}{2!}\right)^3 \Delta t^2}{3!} \\
\boldsymbol{R}^{(2)} &= \boldsymbol{R}^{(0)} - \frac{1}{2}\left(\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t - \frac{2}{2}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2\right) - \frac{1}{6}\left(\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2\right) \\
\boldsymbol{R}^{(2)} &= \boldsymbol{R}^{(0)} - \frac{1}{2}\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t + \frac{1}{3}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2 \qquad \text{(C.14)}
\end{aligned}$$

Finally, for the fourth iteration, $\boldsymbol{R}^{(2)}$ was substituted for $\boldsymbol{R}$ on the right side of (C.11). Then $\boldsymbol{R}^{(3)}$ was found by including all terms with degree less than or equal to Δt^3 from (C.11) and was listed as (C.15). The error of $o(\Delta t^3)$ could be confirmed as before by substituting $\boldsymbol{R}^{(3)}$ from (C.15) into $\boldsymbol{R}$ in (C.10).

$$\begin{aligned}
\boldsymbol{R}^{(3)} &= \boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(2)}\right)^2 \Delta t}{2!} - \frac{\left(\boldsymbol{R}^{(2)}\right)^3 \Delta t^2}{3!} - \frac{\left(\boldsymbol{R}^{(2)}\right)^4 \Delta t^3}{4!} \\
\boldsymbol{R}^{(3)} &= \boldsymbol{R}^{(0)} - \frac{\left(\boldsymbol{R}^{(0)} - \frac{1}{2}\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t + \frac{1}{3}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2\right)^2 \Delta t}{2!} \cdots \\
&\cdots - \frac{\left(\boldsymbol{R}^{(0)} - \frac{1}{2}\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t + \frac{1}{3}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2\right)^3 \Delta t^2}{3!} \cdots \\
&\cdots - \frac{\left(\boldsymbol{R}^{(0)} - \frac{1}{2}\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t + \frac{1}{3}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2\right)^4 \Delta t^3}{4!} \\
\boldsymbol{R}^{(3)} &= \boldsymbol{R}^{(0)} - \frac{1}{2}\left(\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t - \frac{2}{2}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2 + \frac{1}{4}\left(\boldsymbol{R}^{(0)}\right)^4 \Delta t^3 \cdots\right. \\
&\left.\cdots + \frac{2}{3}\left(\boldsymbol{R}^{(0)}\right)^4 \Delta t^3\right) - \frac{1}{6}\left(\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2 - \frac{3}{2}\left(\boldsymbol{R}^{(0)}\right)^4 \Delta t^3\right) \cdots \\
&\cdots - \frac{1}{24}\left(\left(\boldsymbol{R}^{(0)}\right)^4 \Delta t^3\right) \\
\boldsymbol{R}^{(3)} &= \boldsymbol{R}^{(0)} - \frac{1}{2}\left(\boldsymbol{R}^{(0)}\right)^2 \Delta t + \frac{1}{3}\left(\boldsymbol{R}^{(0)}\right)^3 \Delta t^2 - \frac{1}{4}\left(\boldsymbol{R}^{(0)}\right)^4 \Delta t^3 \qquad \text{(C.15)}
\end{aligned}$$

C.3 Derivation: The Baum-Welch algorithm for the k discrete state system with normal emissions

The deivation follows the form suggested in [6] with more explicit steps provided for M-step. The E-step computed $p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta_j})$ and $p(z_{im} = 1, z_{(i+1)n} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta_j})$ iteratively using the forward-backwards algorithm derived in the following section. For efficiency, the following notation was used.

$$\begin{aligned} a_i(m) &= p(z_{im} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta}) & \alpha_i(m) &= p(z_{im} = 1, \boldsymbol{y}_1^{i}|\boldsymbol{\theta}) \\ \beta_i(m) &= p(\boldsymbol{y}_{i+1}^T|z_{im} = 1, \boldsymbol{\theta}) & \gamma_i(m) &= p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}) \\ \Gamma &= p(\boldsymbol{y}_1^T|\boldsymbol{\theta}) & \xi_i(m,n) &= p(z_{in} = 1, z_{(i-1)m} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}) \end{aligned}$$

The M-step follows the derivation of the forward-backward algorithm for this case of HMM. To simplify the computation of the maximums, $\ln(p(\boldsymbol{z}_1^T, \boldsymbol{y}_1^T|\boldsymbol{\theta}))$ was explored. As stated previously, $\boldsymbol{z}_1^T$ was an indicator variable for the state. Therefore, $\mathbb{E}_{\boldsymbol{z}_1^T|\boldsymbol{y}_1^T,\boldsymbol{\theta}_{j-1}}[z_{im} = 1] = p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}) = \gamma_i(m)$ and $\mathbb{E}_{\boldsymbol{z}_1^T|\boldsymbol{y}_1^T,\boldsymbol{\theta}_{j-1}}[z_{im} = 1, z_{(i+1)n} = 1] = p(z_{im} = 1, z_{(i+1)n} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}) = \xi_i(m,n)$. For brevity, the shortened notations above were used along with $\Upsilon_{j-1} = \boldsymbol{z}_1^T|\boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}$ and $\ln \boldsymbol{\mathcal{L}} = \ln(p(\boldsymbol{z}_1^T, \boldsymbol{y}_1^T|\boldsymbol{\theta}))$

$$
\begin{aligned}
\ln \mathcal{L} &= \ln \left\{ p(\boldsymbol{z}_1) \left[\prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i) \right] \left[\prod_{i=1}^{T-1} p(\boldsymbol{z}_{i+1}|\boldsymbol{z}_1) \right] \right\} \\
\ln \mathcal{L} &= \sum_{m=1}^{k} z_{1m} \ln \rho_m + \sum_{i=1}^{T} \sum_{m=1}^{k} z_{im} \ln p(y_i|z_{im}=1) + \ldots \\
&\quad \cdots + \sum_{i=1}^{T-1} \sum_{m=1}^{k} \sum_{n=1}^{k} z_{im} z_{(i+1)n} \ln q_{mn} \\
\mathbb{E}_{\Upsilon_{j-1}}[\ln \mathcal{L}] &= \sum_{m=1}^{k} \gamma_1(m) \ln \rho_m + \sum_{i=1}^{T} \sum_{m=1}^{k} \gamma_i(m) \ln p(y_i|z_{im}=1) \ldots \\
&\quad \cdots + \sum_{i=1}^{T-1} \sum_{m=1}^{k} \sum_{n=1}^{k} \xi_i(m,n) \ln q_{mn} \qquad \text{(C.16)}
\end{aligned}
$$

C.3.1 The forward iteration for the Baum-Welch

On the forward pass, $a_i(m)$ (the state update) and $\alpha_i(m)$ (the state forcast) were computed for $i = 1$ to $i = T$. First, $a_i(m)$ and $\alpha_i(m)$ were computed for $i = 1$ where $a_1(m) = p(z_{1m} = 1|\boldsymbol{\theta})$ because there were no emission prior to t_1.

$$
\begin{aligned}
a_1(m) &= p(z_{1m} = 1|\boldsymbol{\theta}) \\
a_1(m) &= \rho_m \qquad \text{(C.17)} \\
\alpha_1(m) &= p(z_{1m} = 1, y_1|\boldsymbol{\theta}) \\
\alpha_1(m) &= p(y_1|z_{1m} = 1, \boldsymbol{\theta}) p(z_{1m} = 1|\boldsymbol{\theta}) \\
\alpha_1(m) &= p(y_1|z_{1m} = 1, \boldsymbol{\theta}) a_1(k) \qquad \text{(C.18)} \\
y_1|z_{1m} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_m, \sigma^2)
\end{aligned}
$$

Then, the same was done for $i = 1$ to $i = T$.

$$a_i(n) = p(z_{in} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta})$$

$$a_i(n) = \sum_{m=1}^{k} p(z_{in} = 1, z_{(i-1)m} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta})$$

$$a_i(n) = \sum_{m=1}^{k} p(z_{in} = 1|z_{(i-1)m} = 1, \boldsymbol{y}_1^{i-1}, \boldsymbol{\theta})p(z_{(i-1)m} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta})$$

$$a_i(n) = \sum_{m=1}^{k} p(z_{in} = 1|z_{(i-1)m} = 1, \boldsymbol{\theta})\alpha_{i-1}(m)$$

$$a_i(n) = \sum_{m=1}^{k} q_{mn}\alpha_{i-1}(m) \tag{C.19}$$

$$\alpha_i(n) = p(z_{in} = 1, y_1^i|\boldsymbol{\theta})$$

$$\alpha_i(n) = p(y_i|z_{in} = 1, \boldsymbol{\theta})p(z_{in} = 1, \boldsymbol{y}_1^{i-1}|\boldsymbol{\theta})$$

$$\alpha_i(n) = p(y_i|z_{in} = 1, \boldsymbol{\theta})a_i(k) \tag{C.20}$$

$$y_1|z_{in} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_n, \sigma^2)$$

C.3.2 The backward iteration for the Baum-Welch

The backward iteration was constructed such that $\alpha_i(m)\beta_i(m) = p(z_{im} = 1, \boldsymbol{y}_i^T|\boldsymbol{\theta})$. By using the definitions above, this holds for $i = 1$ to $i = T - 1$.

$$\beta_i(m)\alpha_i(m) = p(y_{i+1}^T|z_{im} = 1, \boldsymbol{\theta}_j)p(z_{im} = 1, y_1^i|\boldsymbol{\theta})$$

$$\beta_i(m)\alpha_i(m) = p(y_{i+1}^T|z_{im} = 1, y_1^i, \boldsymbol{\theta})p(z_{im} = 1, y_1^i|\boldsymbol{\theta})$$

$$\beta_i(m)\alpha_i(m) = p(y_{i+1}^T, z_{im} = 1, y_1^i|\boldsymbol{\theta})$$

$$\beta_i(m)\alpha_i(m) = p(z_{im} = 1, y_1^T|\boldsymbol{\theta}) \tag{C.21}$$

However, for the case of $i = T$, the $\alpha_T(n) = p(z_{in} = 1|y_1^T, \boldsymbol{\theta})$ Therefore, to maintain the relation $\alpha_T(n)\beta_T(n) = p(z_{in} = 1, \boldsymbol{y}_i^T|\boldsymbol{\theta})$, it is necessary to set $\beta_T(n) = 1$. Then, for $i = T-1$ to $i = 1$, it is necessary to use equation (C.22).

$$
\begin{aligned}
&\beta_i(n) = p(y_{i+1}^T|z_{im} = 1, \boldsymbol{\theta})\\
&\beta_i(n) = p(y_{i+2}^T, y_{i+1}|z_{im} = 1, \boldsymbol{\theta})\\
&\beta_i(n) = \sum_{n=1}^{k} p(y_{i+2}^T, y_{i+1}, z_{(i+1)n} = 1|z_{im} = 1, \boldsymbol{\theta})\\
&\beta_i(n) = \sum_{n=1}^{k} p(y_{i+2}^T|y_{i+1}, z_{(i+1)n} = 1, z_{im} = 1, \boldsymbol{\theta})p(y_{i+1}, z_{(i+1)m} = 1|z_{in} = 1, \boldsymbol{\theta})\\
&\beta_i(n) = \sum_{n=1}^{k} p(y_{i+2}^T|z_{(i+1)n} = 1, \boldsymbol{\theta})p(y_{i+1}|z_{(i+1)n} = 1, z_{im} = 1, \boldsymbol{\theta}) \times \ldots\\
&\quad \ldots p(z_{(i+1)n} = 1|z_{im} = 1, \boldsymbol{\theta})\\
&\beta_i(n) = \sum_{n=1}^{k} \beta_{i+1}(n)p(y_{i+1}|z_{(i+1)n} = 1, \boldsymbol{\theta})q_{mn}\\
&\beta_i(n) = \sum_{n=1}^{k} q_{mn}p(y_{i+1}|z_{(i+1)n} = 1, \boldsymbol{\theta})\beta_{i+1}(n) \qquad \text{(C.22)}\\
&y_{i+1}|z_{(i+1)m} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_m, \sigma^2)
\end{aligned}
$$

As already shown in equation (C.21), $\alpha_i(m)\beta_i(m) = p(z_{im} = 1, \boldsymbol{y}_1^T|\boldsymbol{\theta})$. However, the two required probabilities are $\gamma_i(m) = p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta})$ and $\xi_i(m,n) = p(z_{im} = 1, z_{(i+1)n} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta})$. $\gamma_i(m)$, comes almost immediately from equation (C.21).

$$
\begin{aligned}
\gamma_i(m) &= p(z_{im} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta})\\
\gamma_i(m) &= \frac{p(z_{im} = 1, \boldsymbol{y}_1^T|\boldsymbol{\theta})}{p(\boldsymbol{y}_1^T|z_{im} = 1)}\\
\gamma_i(m) &= \frac{\alpha_i(m)\beta_i(m)}{\Gamma} \qquad \text{(C.23)}
\end{aligned}
$$

Γ can be found multiple ways using quantities previously calculated. One such way, was to use $\alpha_t(k)$.

$$\begin{aligned}
\Gamma &= p(\boldsymbol{y}_1^T|\boldsymbol{\theta}) \\
\Gamma &= \sum_{m=1}^{k} p(z_{Tm}=1, \boldsymbol{y}_1^T|\boldsymbol{\theta}_j) \\
\Gamma &= \sum_{m=1}^{k} \alpha_T(m) \qquad \text{(C.24)}
\end{aligned}$$

Then, $\gamma_i(m)$ was found by substituting (C.24) into (C.23). The last quantity required was $\xi_i(m,n) = p(z_{im}=1, z_{(i+1)n}=1|\boldsymbol{y}_1^T, \boldsymbol{\theta})$. Consider $\xi_i(m,n)$.

$$\begin{aligned}
&\xi_i(m,n) = p(z_{im}=1, z_{(i+1)n}=1|\boldsymbol{y}_1^T, \boldsymbol{\theta}) \\
&= \frac{p(z_{im}=1, z_{(i+1)n}=1, \boldsymbol{y}_1^T|\boldsymbol{\theta})}{p(\boldsymbol{y}_1^T|\boldsymbol{\theta})} \\
&= \frac{p(z_{im}=1, z_{(i+1)n}=1, \boldsymbol{y}_1^T|\boldsymbol{\theta})}{\Gamma} \\
&= \frac{p(\boldsymbol{y}_1^{i+1}, z_{im}=1, z_{(i+1)n}=1, \boldsymbol{y}_{i+2}^T|\boldsymbol{\theta})}{\Gamma} \\
&= \frac{p(z_{im}=1, z_{(i+1)n}=1, \boldsymbol{y}_1^{i+1}|\boldsymbol{\theta})p(\boldsymbol{y}_{i+2}^T|z_{im}=1, z_{(i+1)n}=1, \boldsymbol{y}_1^{i+1}, \boldsymbol{\theta})}{\Gamma} \\
&= \frac{p(z_{im}=1, z_{(i+1)n}=1, y_{i+1}, \boldsymbol{y}_1^i|\boldsymbol{\theta})p(\boldsymbol{y}_{i+2}^T|z_{(i+1)n}=1, \boldsymbol{\theta})}{\Gamma} \\
&= \frac{p(z_{(i+1)n}=1, y_{i+1}|z_{im}=1, \boldsymbol{y}_1^i, \boldsymbol{\theta})p(z_{im}=1, \boldsymbol{y}_1^i|\boldsymbol{\theta})\beta_{i+1}(n)}{\Gamma} \\
&= \frac{p(z_{im}=1, \boldsymbol{y}_1^i|\boldsymbol{\theta})p(z_{(i+1)n}=1, y_{i+1}|z_{im}=1, \boldsymbol{\theta_j})\beta_{i+1}(n)}{\Gamma} \\
&= \frac{\alpha_i(m)p(y_{i+1}|z_{(i+1)n}=1, z_{im}=1, \boldsymbol{\theta_j})p(z_{(i+1)n}=1|z_{im}=1, \boldsymbol{\theta})\beta_{i+1}(n)}{\Gamma} \\
&= \frac{\alpha_i(m)p(z_{(i+1)n}=1|z_{im}=1, \boldsymbol{\theta})p(y_{i+1}|z_{(i+1)n}=1, \boldsymbol{\theta})\beta_{i+1}(n)}{\Gamma} \\
&\xi_i(m,n) = \frac{\alpha_i(m)q_{mn}p(y_{i+1}|z_{(i+1)n}=1, \boldsymbol{\theta}))\beta_{i+1}(n)}{\Gamma} \qquad \text{(C.25)} \\
&y_{i+1}|z_{(i+1)n}=1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_n, \sigma^2)
\end{aligned}$$

C.3.3 Computing $\hat{\theta}_j$

For this section, the abbreviations from appendix C.3 were used and all derivation started with the expectation of the joint likelihood in (C.16).

C.3.3.1 Deriving $(\hat{\mu}_m)_j$

$$\begin{aligned}
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \mu_m} &= \frac{\partial}{\partial \mu_m 2}\left\{\sum_{m=1}^{k}\gamma_1(m)\ln\rho_m + \sum_{i=1}^{T}\sum_{m=1}^{k}\gamma_i(m)\ln p(y_i|z_{im}=1)\dots\right. \\
&\qquad \left.\dots+\sum_{i=1}^{T-1}\sum_{m=1}^{k}\sum_{n=1}^{k}\xi_i(m,n)\ln q_{mn}\right\} \\
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \mu_m} &= \frac{\partial}{\partial \mu_m}\left\{\sum_{i=1}^{T}\gamma_i(m)\ln p(y_i|z_{im}=1)\right\} \\
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \mu_m} &= \frac{\partial}{\partial \mu_m}\left\{\sum_{i=1}^{T}\gamma_i(m)\frac{(y_i-\mu_m)^2}{-2\sigma^2}\right\} \\
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \mu_m} &= \frac{\partial}{\partial \mu_m}\left\{\sum_{i=1}^{T}\frac{(\gamma_i(m)y_i^2-2\gamma_i(m)y_i\mu_m+\gamma_i(m)\mu_m^2)}{2\sigma^2}\right\} \\
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \mu_m} &= \sum_{i=1}^{T}\frac{(2\gamma_i(m)y_i-2\gamma_i(m)y_i\mu_m)}{-2\sigma^2} \\
(\hat{\mu}_m)_j &= \frac{\sum_{i=1}^{T}(\gamma_i(m)y_i)}{\sum_{i=1}^{T}\gamma_i(m)} \qquad \text{(C.26)}
\end{aligned}$$

C.3.3.2 Deriving $\hat{\sigma}_j^2$

$$\begin{aligned}
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \sigma^2} &= \frac{\partial}{\partial \sigma^2}\left\{\sum_{m=1}^{k}\gamma_1(m)\ln\rho_m + \sum_{i=1}^{T}\sum_{m=1}^{k}\gamma_i(m)\ln p(y_i|z_{im}=1)\dots\right. \\
&\qquad \left.\dots+\sum_{i=1}^{T-1}\sum_{m=1}^{k}\sum_{n=1}^{k}\xi_i(m,n)\ln q_{mn}\right\} \\
\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \sigma^2} &= \sum_{i=1}^{T}\sum_{m=1}^{k}\gamma_i(m)\ln p(y_i|z_{im}=1)
\end{aligned}$$

$$\frac{\partial \mathbb{E}_{\Upsilon_j}[\ln \mathcal{L}]}{\partial \sigma^2} = -\frac{T}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{T}\sum_{m=1}^{k}\left[\gamma_i(m)\left(y_i - (\mu_m)_j\right)^2\right]$$

$$\hat{\sigma}_j^2 = \frac{\sum\limits_{i=1}^{T}\sum\limits_{m=1}^{k}\left[\gamma_i(m)\left(y_i - (\mu_m)_j\right)^2\right]}{T} \tag{C.27}$$

C.3.3.3 Deriving $(\hat{q}_{mn})_j$

q_{mn} was only present in part of the likelihood, therefore it was sufficient to find the each q_{mn} that maximized $\sum\limits_{i=1}^{T-1}\sum\limits_{m=1}^{k}\sum\limits_{n=1}^{k}\xi_i(m,n)\ln q_{mn}$. Since q_{mn} was first degree, the standard method of differentiating the likelihood function was not helpful in finding the q_{mn} that maximized $\ln \mathcal{L}$. Therefore, q_{mn} where m was fixed and $n = 1$ to k was considered. Since this represents all transitions from state m, then the sum of all q_{mn} where m was fixed and $n = 1$ to k was one. This resulted in a constrained optimization problem that could be written as a Lagrange function in (C.28).

$$L(q_{mn}, \lambda) = \left[\sum_{i=1}^{T}\sum_{n=1}^{k}\ln q_{mn}\xi_i(m,n)\right] - \lambda\left[\left(\sum_{n=1}^{k} e^{\ln q_{mn}}\right) - 1\right] \tag{C.28}$$

Then $(\hat{q}_{mn})_j$ was found by differentiating and solving. (C.29) was found by differentiating with respect to $\ln q_{mh}$ where h was fixed.

$$\begin{aligned}
\frac{\partial L(q_{mn},\lambda)}{\partial \ln q_{mh}} &= \frac{\partial}{\partial \ln q_{mh}}\left\{\left[\sum_{i=1}^{T}\sum_{n=1}^{k}\ln q_{mn}\xi_i(m,n)\right]-\lambda\left[\left(\sum_{n=1}^{k}e^{\ln q_{mn}}\right)-1\right]\right\}\\
\frac{\partial L(q_{mn},\lambda)}{\partial \ln q_{mh}} &= \left[\sum_{i=1}^{T}\xi_i(m,h)\right]-\lambda e^{\ln q_{mh}}\\
0 &= \left[\sum_{i=1}^{T}\xi_i(m,h)\right]-\hat{\lambda}_j\left(\hat{q}_{mh}\right)_j\\
\left(\hat{q}_{mh}\right)_j &= \frac{\sum_{i=1}^{T}\xi_i(m,h)}{\hat{\lambda}_j} \qquad \text{(C.29)}
\end{aligned}$$

To find $\left(q_{mh}\right)_j$, it was necessary to find the value of $\hat{\lambda}_j$ in (C.29). Therefore $L(q_{mn},\lambda)$ was differentiated in terms of λ, resulting in a second equation listed as (C.30).

$$\begin{aligned}
\frac{\partial L(q_{mn},\lambda)}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\left\{\sum_{n=1}^{k}\ln q_{mn}\xi_i(m,n)-\lambda\left[\left(\sum_{n=1}^{k}e^{\ln q_{mn}}\right)-1\right]\right\}\\
\frac{\partial L(q_{mn},\lambda)}{\partial \lambda} &= \left(\sum_{n=1}^{k}e^{\ln q_{mn}}\right)-1\\
0 &= \left(\sum_{n=1}^{k}q_{mn}\right)-1\\
\left(\sum_{n=1}^{k}q_{mn}\right) &= 1 \qquad \text{(C.30)}
\end{aligned}$$

Since h in (C.29) was general, (C.29) could be used for any $h \in \{1,2,...,k\}$. Therefore, in (C.31), the sum of q_{mn} for all $n \in \{1,2,...,k\}$ was put on the left and the sum of the right side of (C.29) for all $n \in \{1,2,...,k\}$ was put on the right.

$$\sum_{n=1}^{k}\left(\hat{q}_{mn}\right)_j = \sum_{n=1}^{k}\frac{\sum_{i=1}^{T}\xi_i(m,n)}{\hat{\lambda}_j} \qquad \text{(C.31)}$$

It was known from (C.30) that $\left(\sum_{n=1}^{k} q_{mn}\right) = 1$. This was substituted into (C.31) and then $\hat{\lambda}_j$ that maximized $L(q_{mn}, \lambda)$ was found.

$$\begin{aligned}
\sum_{n=1}^{k} (\hat{q}_{mn})_j &= \sum_{n=1}^{k} \frac{\sum_{i=1}^{T} \xi_i(m,n)}{\hat{\lambda}_j} \\
1 &= \frac{\sum_{n=1}^{k} \sum_{i=1}^{T} \xi_i(m,n)}{\hat{\lambda}_J}
\end{aligned}$$

Since the summation above is finite, the order of the summations can be changed.

$$\begin{aligned}
\hat{\lambda}_j &= \sum_{i=1}^{T} \sum_{n=1}^{k} \xi_i(m,n) \\
\hat{\lambda}_j &= \sum_{i=1}^{T} \sum_{n=1}^{k} p(z_{im} = 1, z_{(i+1)n} = 1 | \boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}) \qquad \text{(C.32)}
\end{aligned}$$

Finally, applying the law of total probability to (C.32), $\sum_{n=1}^{k} p(z_{im} = 1, z_{(i+1)n} = 1 | \boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}) = p(z_{im} = 1 | \boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1}) = \gamma_i(m)$. This resulted in $(\hat{q}_{mn})_j$ listed as (C.33)

$$\begin{aligned}
(\hat{q}_{mh})_j &= \frac{\sum_{i=1}^{T} \xi_i(m,h)}{\sum_{i=1}^{T} \sum_{n=1}^{k} p(z_{im} = 1, z_{(i+1)n} = 1 | \boldsymbol{y}_1^T, \boldsymbol{\theta}_{j-1})} \\
(\hat{q}_{mh})_j &= \frac{\sum_{i=1}^{T} \xi_i(m,h)}{\sum_{i=1}^{T} \gamma_i(m)} \qquad \text{(C.33)}
\end{aligned}$$

C.3.3.4 Deriving $(\hat{\rho}_m)_j$

Finding $(\hat{\rho_m})_j$ presents the many of the same problems as computing $(\hat{q}_{mn})_j$. Furthermore, $\sum_{m=1}^{k} \rho_m = 1$ as with $(\hat{q}_{mn})_j$. Therefore, the derivation of $(\rho_m)_j$ was very similar to $(\hat{q}_{mn})_j$. Following that, the explanations in this section were abbreviated. See appendix C.3.3.3 for a more thorough explanations of the steps.

The only terms that contained ρ_m were the terms $\sum_{m=1}^{k} \gamma_1(m) \ln \rho_m$. Therefore, to maximize the likelihood in terms of ρ_m, it was sufficient to maximize

$\sum\limits_{m=1}^{k} \gamma_1(m) \ln \rho_m$ for ρ_h where h was fixed. First, $(\hat{\rho}_h)_j$ was solved in terms of $\hat{\lambda}_j$.

$$\begin{aligned}
L(\rho_m, \lambda) &= \sum_{m=1}^{k} \ln \rho_m \gamma_1(m) - \lambda \left[\left(\sum_{m=1}^{k} e^{\ln \rho_m} \right) - 1 \right] \\
\frac{\partial L(\boldsymbol{\rho}, \lambda)}{\partial \ln \rho_h} &= \frac{\partial}{\partial \ln \rho_h} \left\{ \sum_{m=1}^{k} \ln \rho_m \gamma_1(m) - \lambda \left[\left(\sum_{m=1}^{k} e^{\ln \rho_m} \right) - 1 \right] \right\} \\
\frac{\partial L(\boldsymbol{\rho}, \lambda)}{\partial \ln \rho_h} &= [\gamma_1(h)] - \lambda e^{\ln \rho_h} \\
0 &= \gamma_1(h) - \hat{\lambda}_j \left(\hat{\rho}_h\right)_j \\
\left(\hat{\rho}_h\right)_j &= \frac{\gamma_1(h)}{\hat{\lambda}_j} \qquad \text{(C.34)}
\end{aligned}$$

Using the constraint $\sum_{m=1}^{k} \rho_m = 1$, λ was solved for.

$$\begin{aligned}
\sum_{m=1}^{k} \rho_m &= \sum_{m=1}^{k} \frac{\gamma_1(m)}{\lambda} \\
1 &= \frac{\sum_{m=1}^{k} \gamma_1(m)}{\lambda} \\
\lambda &= \sum_{m=1}^{k} \gamma_1(m) \qquad \text{(C.35)}
\end{aligned}$$

(C.34) and (C.35) were combined and $\left(\hat{\rho_m}\right)_j$ was listed as (C.36).

$$\left(\hat{\rho}_h\right)_j = \frac{\gamma_1(h)}{\sum_{m=1}^{k} \gamma_1(m)} \qquad \text{(C.36)}$$

C.4 Derivation: The gibbs sampler for the k discrete state system with k distinct normal emissions

The Gibbs sampler for the k discrete state model with k distinct Gaussian emissions made inference on the probability density function listed as (C.37).

(C.37) $p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})$ for k states

$$\begin{aligned}
p(\boldsymbol{y}_1^T \boldsymbol{z}_1^T|\boldsymbol{\theta}) &= \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) &= \prod_{j=1}^{k} \left[p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1} = j)\right]^{\boldsymbol{z}_{(i-1)j}} \\
p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1} = j) &= \prod_{l=1}^{k} q_{jl}^{z_{il}} \\
p(\boldsymbol{z}_1|\boldsymbol{\theta}) &= \prod_{j=1}^{k} \rho_j^{z_{1j}} \\
y_i|\boldsymbol{z}_{ij} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_j, \sigma^2) \\
p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) &= \prod_{j=1}^{k} \left[p(y_i|\boldsymbol{z}_{ij} = 1, \boldsymbol{\theta})\right]^{z_{ij}}
\end{aligned}$$

Unlike the Baum-Welch and Viterbi training, the form of (C.37) was more convenient than the logarithm of the likelihood for the following derivations. Therefore, all derivations used the form in (C.37).

C.4.1 Derivation: The forward filter backwards sample algorithm for k discrete states with k distinct Gaussian emissions

C.4.1.1 Derivation: The forward filter

The state update and state forecast were defined as in (C.38) and (C.39).

$$a_i(m) = p(z_{im} = 1 | \boldsymbol{y}_1^{i-1}, \boldsymbol{\theta}) \tag{C.38}$$

$$\alpha_i(m) = p(z_{im} = 1 | \boldsymbol{y}_1^{i}, \boldsymbol{\theta}) \tag{C.39}$$

On the forward pass, the state update and the state forecast were computed for $i = 0$ to $i = T$. First, $a_i(k)$ and $\alpha_i(k)$ were computed for $i = 0$.

$$\begin{aligned}
a_1(m) &= p(z_{1m} = 1 | \boldsymbol{\theta}) \\
a_1(m) &= \rho_m && \text{(C.40)} \\
\alpha_1(m) &= p(z_{1m} = 1 | \boldsymbol{y}_1, \boldsymbol{\theta}) \\
\alpha_1(m) &= \frac{p(y_1 | z_{im} = 1, \boldsymbol{\theta}) p(z_{im} = 1 | y_1, \boldsymbol{\theta})}{p(y_1 | \boldsymbol{\theta})} \\
\alpha_1(m) &= \frac{p(y_1 | z_{im} = 1, \boldsymbol{\theta}) a_1(m)}{\sum_{n=1}^{k} p(y_1 | z_{in} = 1, \boldsymbol{\theta}) p(z_{in} = 1)} \\
\alpha_1(m) &= \frac{p(y_1 | z_{im} = 1, \boldsymbol{\theta}) a_1(m)}{\sum_{n=1}^{k} p(y_1 | z_{in} = 1, \boldsymbol{\theta}) a_0(n)} && \text{(C.41)} \\
y_1 | z_{1n} = 1, \boldsymbol{\theta} &\sim \mathcal{N}(\mu_n, \sigma^2)
\end{aligned}$$

Then, the same was done for $i = 1$ to $i = T$.

$$a_i(n) = p(z_{in} = 1|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta})$$
$$a_i(n) = \sum_{m=1}^{k} p(z_{in} = 1|z_{(i-1)m} = 1, \boldsymbol{y}_1^{i-1}, \boldsymbol{\theta})p(z_{(i-1)m} = 1|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta})$$

Since $p(z_{in} = 1|z_{(i-1)m} = 1, \boldsymbol{y}_1^{i-1}, \boldsymbol{\theta})$ is conditionally independent of $\boldsymbol{y}_1^{i-1}$ and $p(z_{(i-1)m} = 1|\boldsymbol{y}_1^{i-1}, \boldsymbol{\theta}) = \alpha_{i-1}(m)$, $a_i(n)$ was simplified.

$$a_i(n) = \sum_{m=1}^{k} p(z_{in} = 1|z_{(i-1)m} = 1, \boldsymbol{\theta})\alpha_{i-1}(m)$$
$$a_i(n) = \sum_{m=1}^{k} \alpha_{i-1}(m)q_{mn} \tag{C.42}$$

Finally, C.41 generalizes to the case of $\alpha_i(m)$.

$$\alpha_i(n) = \frac{p(y_i|z_{in} = 1, \boldsymbol{\theta})a_i(n)}{\sum_{m=1}^{k} p(y_i|z_{im} = 1, \boldsymbol{\theta})a_i(m)} \tag{C.43}$$
$$y_i|z_{im} = 1, \boldsymbol{\theta} \sim \mathcal{N}(\mu_m, \sigma^2)$$

C.4.1.2 Derivation: Backwards Sampling

Once the state update has been calculated for $t = 0$ to $t = T$, $\boldsymbol{z}_i|\boldsymbol{y}_1^T, \boldsymbol{\theta}$ was drawn from $i = T$ to 1. Since there was no information for the time t_{T+1}, $\boldsymbol{z}_T$ was drawn by calculating $p(\boldsymbol{z}_T|\boldsymbol{y}_1^T, \boldsymbol{\theta})$. This comes immediately from prior calculations. For $i = T$, $\alpha_T(m) = p(z_{Tm} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta})$, so $\boldsymbol{z}_T \sim Cat(\alpha_T(1), \alpha_T(2), ..., \alpha_T(k))$ where Cat denotes the categorical distribution. Then for all $i < T$, $\boldsymbol{z}_{i+1}$ was known prior to drawing $\boldsymbol{z}_i$ so it was sufficient to calculate $p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^T, \boldsymbol{\theta}) = p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{y}_{i+1}^T, \boldsymbol{\theta})$ to draw $\boldsymbol{z}_i$. Since $\boldsymbol{z}_i$ couldn't be directly drawn from $p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^T, \boldsymbol{\theta})$ Bayes' rule was applied to $p(z_{im} =$

$1|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{y}_{i+1}^T, \boldsymbol{\theta})$ resulting in $\frac{p(\boldsymbol{y}_{i+1}^T|z_{im}=1,z_{(i+1)n}=1,\boldsymbol{y}_1^i,\boldsymbol{\theta})p(z_{im}=1|z_{(i+1)n}=1,\boldsymbol{y}_1^i,\boldsymbol{\theta})}{p(\boldsymbol{y}_{i+1}^T|z_{(i+1)n}=1,\boldsymbol{y}_1^i,\boldsymbol{\theta})}$. Following this, $p(\boldsymbol{y}_{i+1}^T|z_{im} = 1, z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta})$ is conditionally independent of $z_{im} = 1$ and $\boldsymbol{y}_1^i$, therefore the expression was rewritten and then simplified further below.

$$\begin{aligned}
p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^T, \boldsymbol{\theta}) &= \frac{p(\boldsymbol{y}_{i+1}^T|z_{(i+1)n} = 1, \boldsymbol{\theta})p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta})}{p(\boldsymbol{y}_{i+1}^T|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta})} \\
&\propto p(\boldsymbol{y}_{i+1}^T|z_{(i+1)n} = 1, \boldsymbol{\theta})p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta}) \\
&\propto p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta}) \\
&\propto \frac{p(z_{(i+1)n} = 1|z_{im} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta})p(z_{im} = 1|\boldsymbol{y}_1^i, \boldsymbol{\theta})}{p(z_{(i+1)n} = 1|\boldsymbol{y}_1^T, \boldsymbol{\theta})} \\
&\propto p(z_{(i+1)n} = 1|z_{im} = 1, \boldsymbol{y}_1^i, \boldsymbol{\theta})p(z_{im} = 1|\boldsymbol{y}_1^i, \boldsymbol{\theta}) \\
&\propto p(z_{(i+1)n} = 1|z_{im} = 1, \boldsymbol{\theta})p(z_{im} = 1|\boldsymbol{y}_1^i, \boldsymbol{\theta}) \qquad \text{(C.44)}
\end{aligned}$$

Since $p(z_{im} = 1|\boldsymbol{y}_1^t, \boldsymbol{\theta}) = \alpha_i(m)$ equation (C.44) was re-written in terms of $\alpha_i(m)$.

$$p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^T, \boldsymbol{\theta}) \propto \alpha_i(m)q_{mn}$$

Then $p(z_{im} = 1|z_{(i+1)n} = 1, \boldsymbol{y}_1^T, \boldsymbol{\theta})$ can be written as $p(z_{im} = 1|z_{(i+1)n}, \boldsymbol{y}_1^T, \boldsymbol{\theta}) \propto \sum_{n=1}^{k} \alpha_i(m)q_{mn}z_{(i+1)n}$. Since $\boldsymbol{z}_i$, was an indicator variable it is not difficult to normalize and express $p(z_{im} = 1|\boldsymbol{z}_{(i+1)n}, \boldsymbol{y}_1^T, \boldsymbol{\theta})$ as a proper distribution. For this, the definition in C.45 was used to compute $\beta_i(m)$ in C.46 which was used in the distribution to draw $\boldsymbol{z}_i$ listed as (C.47).

$$B_i(m) = \sum_{n=1}^{k} \alpha_i(m) q_{mn} z_{(i+1)n} \tag{C.45}$$

$$\beta_i(m) = \frac{B_i(m)}{\sum_{n=1}^{k} B_i(n)} \tag{C.46}$$

$$\boldsymbol{z}_i \sim Cat(\beta_i(1), \beta_i(2), ..., \beta_i(k)) \tag{C.47}$$

C.4.2 Derivation: Sampling $\boldsymbol{\theta}$ for k discrete states with k distinct Gaussian emissions

C.4.2.1 Derivation: Sampling the conditional posterior of $\boldsymbol{\rho}$

Since the prior was conditionally independent, $p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta})$ could be written as a product of the joint likelihood and prior. Given that $p(\boldsymbol{z}_1|\boldsymbol{\Theta})$ is a draw from a categorical distribution the conditionally conjugate prior is $\boldsymbol{\rho} \sim Dir_(\kappa_{\rho_1}, \kappa_{\rho_2}, \ldots, \kappa_{\rho_k})$.

$$\begin{aligned}
p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho}) &\propto p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})p(\boldsymbol{\rho}) \\
p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho}) &\propto p(\boldsymbol{z}_1|\boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})p(\boldsymbol{\rho}) \\
p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho}) &\propto p(\boldsymbol{z}_1|\boldsymbol{\Theta})p(\rho) \\
p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho}) &\propto \rho_1^{z_{11}} \rho_2^{z_{12}} \ldots \rho_k^{z_{1k}} \rho_1^{(\kappa_{\rho_1}-1)} \rho_2^{(\kappa_{\rho_2}-1)} \ldots \rho_k^{(\kappa_{\rho_k}-1)} \\
p(\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho}) &\propto \rho_1^{(z_{11}+\kappa_{\rho_1}-1)} \rho_2^{(z_{12}+\kappa_{\rho_2}-1)} \ldots \rho_k^{(z_{1k}+\kappa_{\rho_k}-1)} \\
\boldsymbol{\rho}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\rho} &\sim Dir_{\boldsymbol{\rho}}(z_{11} + \kappa_{\rho_1}, z_{12} + \kappa_{\rho_2}, ..., z_{1k} + \kappa_{\rho_k})
\end{aligned} \tag{C.48}$$

C.4.2.2 Derivation: Sampling the conditional posterior of $\boldsymbol{q}_{m\text{-}}$

Since a conditionally independent conjugate prior was chosen, the conditional posterior could be written as a product of the joint likelihood and the prior. Recall, the conditionally conjugate prior was $\boldsymbol{q}_{m\text{-}} \sim Dir(\kappa_{q_{m1}}, \kappa_{q_{m2}}, \ldots, \kappa_{q_{mk}})$ and $\boldsymbol{q}_{m\text{-}} = (q_{m1}, q_{m2}, \ldots, q_{mk})$.

$$
\begin{aligned}
p(\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}}) &\propto p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})p(\boldsymbol{q}_{m\text{-}}) \\
p(\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}}) &\propto p(\boldsymbol{z}_1|\boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})p(\boldsymbol{q}_{m\text{-}}) \\
p(\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}}) &\propto \left[\prod_{i=1}^{T} p(\boldsymbol{z}_i|z_{(i-1)1} = 1, \boldsymbol{\Theta})\right] p(\boldsymbol{q}_{m\text{-}}) \\
p(\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}}) &\propto \left\{\prod_{i=1}^{T} \left[\prod_{n=1}^{k} q_{mn}^{z_{in}}\right]^{z_{(i-1)m}}\right\} \prod_{n=1}^{k} q_{mn}^{\kappa_{q_{mn}}} \\
p(\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}}) &\propto \prod_{n=1}^{k} q_{mn}^{\left(\sum_{i=1}^{T} z_{in}z_{(i-1)m} + \kappa_{q_{mn}}\right)} \\
\boldsymbol{q}_{m\text{-}}|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\boldsymbol{q}_{m\text{-}}} &\sim Dir\left(\sum_{i=1}^{T} z_{i1}z_{(i-1)m} + \kappa_{q_{m2}}, ..., \sum_{i=1}^{T} z_{ik}z_{(i-1)m} + \kappa_{q_{mk}}\right)
\end{aligned}
\tag{C.49}
$$

C.4.2.3 Derivation: Sampling the conditional posterior of μ_n

As with the other parameters, the the conditionally conjugate prior was used. For the case of μ_n, the conditionally independent conjugate prior for μ_n was $\mathcal{N}(m_n, \sigma^2)$. Since the conditionally conjugate prior was independent, the conditional posterior was proportional to the product of the joint likelihood and the conditional prior.

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto p(\boldsymbol{z}_1|\boldsymbol{\theta}) \prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) \prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta}) p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \left[\prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})\right] p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \left[\prod_{i=1}^{T} \left\{ \prod_{m=1}^{k} p(y_i|z_{im} = 1, \boldsymbol{\theta})^{z_{im}} \right\}\right] p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \left[\prod_{i=1}^{T} p(y_i|z_{im} = 1, \boldsymbol{\theta})^{z_{im}}\right] p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \left[\prod_{i=1}^{T} \exp\left\{ \frac{1}{2\sigma^2} \left[(y_i - \mu_n)^2(z_{in})\right] \right\}\right] p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i=1}^{T}(y_i - \mu_n)^2(z_{in})\right] \right\} p(\mu_n)$$

$$p(\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n}) \propto \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i=1}^{T}((z_{in})y_i^2 - 2(z_{in})y_i\mu_n + (z_{in})\mu_n^2)\right] \right\} p(\mu_n)$$

$$\propto \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i=1}^{T}((z_{in})y_i^2 - 2(z_{in})y_i\mu_n + (z_{in})\mu_n^2))\right] \right\} p(\mu_n)$$

$$\propto \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i=1}^{T}(-2(z_{in})y_i\mu_n + (z_{in})\mu_n^2)\right] - \frac{1}{2\sigma_n^2}\left[\mu_n^2 - 2\mu_n m_n\right] \right\}$$

$$\propto \exp\left\{ \frac{-1}{2\sigma^2\sigma_n^2} \left[\mu_n^2(\sigma_n^2 \sum_{i=0}^{T}(z_{in}) + \sigma^2 m_n) - 2\mu_n(\sigma_n^2 \sum_{i=1}^{T}(z_{in})y_i + \sigma^2 m_n)\right] \right\}$$

$$\propto \exp\left\{ -\frac{\sigma_n^2 \sum_{i=0}^{T}(z_{in}) + \sigma^2 m_n}{2\sigma^2\sigma_n^2} \left[\mu_n^2 - 2\mu_n \frac{\sigma_n^2 \sum_{i=1}^{T}(z_{in})y_i + \sigma^2 m_n}{\sigma_n^2 \sum_{i=0}^{T}(z_{in}) + \sigma^2 m_{z_{in}}}\right] \right\}$$

$$\propto \exp\left\{ -\frac{\sigma_n^2 \sum_{i=1}^{T}(z_{in}) + \sigma^2 m_n}{2\sigma^2\sigma_n^2} \left[\mu_n^2 - \frac{\sigma_n^2 \sum_{i=1}^{T}(z_{in})y_i + \sigma^2 m_n}{\sigma_n^2 \sum_{i=1}^{T}(z_{in}) + \sigma^2 m_n}\right]^2 \right\}$$

The result above was proportional to a normal distribution resulting in (C.50)

$$\mu_n|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\mu_n} \sim \mathcal{N}\left(\frac{\sigma_n^2 \sum_{i=1}^{T}(z_{in})y_i + \sigma^2 m_n}{\sigma_n^2 \sum_{i=1}^{T}(z_{in}) + \sigma^2 m_n}, \frac{\sigma^2\sigma_n^2}{\sigma_n^2 \sum_{i=1}^{T}(z_{in}) + \sigma^2 m_n} \right) \qquad \text{(C.50)}$$

C.4.2.4 Derivation: Sampling the conditional posterior of σ^2

For σ^2, the independent conditionally conjugate prior was $\sigma^2 \sim \mathcal{IG}\left(\frac{n_{\sigma 2}}{2}, \frac{d_{\sigma 2}}{2}\right)$. As with the aforementioned conditional posteriors, the independence of the conditional prior makes $p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2})$ proportional to the joint likelihood.

$$
\begin{aligned}
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto p(\boldsymbol{y}_1^T, \boldsymbol{z}_1^T|\boldsymbol{\theta})p(\sigma^2)\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto p(\boldsymbol{z}_1|\boldsymbol{\theta})\prod_{i=2}^{T} p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1})\prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})p(\sigma^2)\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto \left[\prod_{i=1}^{T} p(y_i|\boldsymbol{z}_i, \boldsymbol{\theta})\right] p(\sigma^2)\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto \left[\prod_{i=1}^{T}\left\{\prod_{m=1}^{k} p(y_i|z_{im}=1, \boldsymbol{\theta})^{z_{im}}\right\}\right] p(\sigma^2)
\end{aligned}
$$

For brevity, let $S_i^2 = \sum\limits_{m=1}^{k}(y_i - \mu_m)^2(z_{im})$ and $S^2 = \sum\limits_{i=1}^{T} S_i^2$. Also, since the system can only be in one state at time t_i, $\sum_{m=1}^{k} z_{im} = 1$ for all i.

$$
\begin{aligned}
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto \left[\prod_{i=0}^{T}(\sigma^2)^{\frac{\sum_{m=1}^{k} z_{im}}{-2}}\exp\left\{\frac{S_i^2}{-2\sigma^2}\right\}\right] p(\sigma^2)\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto (\sigma^2)^{-\frac{T}{2}}\exp\left\{-\frac{\sum_{i=0}^{T} S_i^2}{2\sigma^2}\right\} p(\sigma^2)\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto (\sigma^2)^{-\frac{T}{2}}\exp\left\{-\frac{S^2}{2\sigma^2}\right\}(\sigma^2)^{-\frac{n_{\sigma 2}}{2}-1}\exp\left\{-\frac{d_{\sigma^2}}{2\sigma^2}\right\}\\
p(\sigma^2|\boldsymbol{z}_1^T, \boldsymbol{y}_1^T, \boldsymbol{\theta}_{-\sigma^2}) &\propto (\sigma^2)^{-\frac{T+n_{\sigma 2}}{2}-1}\exp\left\{-\frac{S^2+d_{\sigma^2}}{2\sigma^2}\right\}\\
\sigma^2|\boldsymbol{y}_1^T, \boldsymbol{z}_1^T, \boldsymbol{\theta}_{-\sigma^2} &\sim \mathcal{IG}\left(\frac{n_{\sigma^2}+T}{2}, \frac{d_{\sigma^2}+S^2}{2}\right) \qquad \text{(C.51)}
\end{aligned}
$$

Appendix D

Supplement: Modeling of Biochemical States of DNA Replication Restricted to Three States

D.1 Derivation: The Gaussian process regression

D.1.1 Derivation of the conditional posterior of $s^2|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-s^2}$ for the Gaussian process regression

Before deriving the posterior, $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\vartheta})$ was simplified in terms of $\boldsymbol{S}^2$ as in (D.1).

$$
\begin{aligned}
\boldsymbol{S}^2 &= (\boldsymbol{y}-\boldsymbol{X\beta})^T \left(\boldsymbol{C}+\nu^2 I\right)^{-1}(\boldsymbol{y}-\boldsymbol{X\beta}) \qquad \text{(D.1)}\\
p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta}) &\propto |s^2\boldsymbol{C}+s^2\nu^2 I|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left((\boldsymbol{y}-\boldsymbol{X\beta})^T\left(s^2\boldsymbol{C}+s^2\nu^2 I\right)^{-1}(\boldsymbol{y}-\boldsymbol{X\beta})\right)\right\}\\
p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta}) &\propto |s^2\boldsymbol{C}+s^2\nu^2 I|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2s^2}\left((\boldsymbol{y}-\boldsymbol{X\beta})^T\left(\boldsymbol{C}+\nu^2 I\right)^{-1}(\boldsymbol{y}-\boldsymbol{X\beta})\right)\right\}\\
p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta}) &\propto |s^2\boldsymbol{C}+s^2\nu^2 I|^{-\frac{1}{2}} \exp\left\{-\frac{\boldsymbol{S}^2}{2s^2}\right\}
\end{aligned}
$$

Two conditional priors were discussed for s^2, the inverse gamma and the truncated inverse gamma distribution. First, the conditional posterior listed as (D.3) was derived when the conditional prior was an inverse gamma distribution as in (D.2). As in section 4.2.4, $\boldsymbol{\vartheta}_{-a}$ denotes all parameters except a.

$$
\begin{aligned}
s^2 &\sim \mathcal{IG}\left(\frac{n_{s^2}}{2},\frac{d_{s^2}}{2}\right) \qquad \text{(D.2)}\\
p(s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2}) &= \frac{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta})p(s^2)}{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2})}\\
p(s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2}) &\propto p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\vartheta})p(s^2)\\
p(s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2}) &\propto |s^2\boldsymbol{C}+s^2\nu^2 I|^{-\frac{1}{2}} \exp\left\{-\frac{\boldsymbol{S}^2}{2s^2}\right\}(s^2)^{-\frac{n_{s2}}{2}-1}\exp\left\{-\frac{d_{s^2}}{2s^2}\right\}\\
p(s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2}) &\propto (s^2)^{-\frac{n_{s2}}{2}} \exp\left\{-\frac{\boldsymbol{S}^2}{2s^2}\right\}(s^2)^{-\frac{n_{s2}}{2}-1}\exp\left\{-\frac{d_{s^2}}{2s^2}\right\}\\
p(s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2}) &\propto (s^2)^{\frac{-n_{s2}-N}{2}-1}\exp\left\{-\frac{1}{2s^2}\left[\boldsymbol{S}^2+d_{s^2}\right]\right\}\\
s^2|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-s^2} &\sim \mathcal{IG}\left(\frac{n_{s^2}+N}{2},\frac{1}{2}\left(\boldsymbol{S}^2+d_{s^2}\right)\right) \qquad \text{(D.3)}
\end{aligned}
$$

As discussed earlier, the inverse gamma is proportional to the truncated inverse gamma with a maximum of m when $0 < s^2 \leq m$. Therefore, the probability function of the conditional posterior when the prior was the truncated inverse gamma was proportional to the conditional posterior when the prior is inverse gamma when $0 < s^2 \leq m$. Therefore, the posterior when the prior was the truncated inverse gamma was simply a truncated inverse gamma distribution with the parameters equal to (D.3). This was listed as (D.4)

$$\begin{aligned} s^2 &\sim \mathcal{TIG}\left(\frac{n_{s^2}}{2}, \frac{d_{s^2}}{2}, m\right) \\ s^2|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-s^2} &\sim \mathcal{TIG}\left(\frac{n_{s^2}+N}{2}, \frac{1}{2}\left(\boldsymbol{S}^2 + d_{s^2}\right), m\right) \end{aligned} \tag{D.4}$$

D.1.2 Derivation of the conditional posterior of $\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\beta}$ for the Gaussian process regression

The derivation here assumes the flat prior listed in (D.5). For discussion of numerical issues to sampling the conditional posterior, see section (4.2.4). As with section (4.2.4), $s^2\boldsymbol{C} + s^2\nu^2\boldsymbol{I}$ was represented by $\boldsymbol{\Sigma}_{GP}$

$$p(\boldsymbol{\vartheta}) \propto 1 \tag{D.5}$$

$$p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\vartheta})p(\boldsymbol{\beta})}{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta})}$$

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta}) &\propto p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\vartheta})p(\boldsymbol{\beta}) \\ p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta}) &\propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right)^T \boldsymbol{\Sigma}_{GP}^{-1}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right)\right\} \times 1 \\ p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta}) &\propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{y}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X\beta} + \cdots\right.\right. \\ &\qquad \left.\left.\cdots - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{XB}\right)\right\} \\ p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\vartheta}_{-\beta}) &\propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{y}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1} \times \cdots\right.\right. \\ &\qquad \left.\left.\cdots \times \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right)\right\} \end{aligned}$$

$$
\begin{aligned}
p(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-\beta}) \propto \exp\Bigg\{-\frac{1}{2}\left(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta}\right. \cdots \\
\cdots - \left[\left(\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\right)^T\boldsymbol{X}^T\left(\boldsymbol{\Sigma}_{GP}^{-1}\right)^T\boldsymbol{y}\right]^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta}\cdots \\
\cdots -\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]\left.\right)\Bigg\}
\end{aligned}
$$

Since $\boldsymbol{\Sigma}_{GP}$ was an invertible symmetric matrix, $\boldsymbol{\Sigma}_{GP}^{-1}$ was also symmetric.

$$
\begin{aligned}
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \left[\left(\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^T\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]^T \times \cdots \\
&\cdots\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]\Bigg)\Bigg\} \\
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \left[\left(\boldsymbol{X}^T\left(\boldsymbol{\Sigma}_{GP}^{-1}\right)^T\left(\boldsymbol{X}^T\right)^T\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]^T \cdots \\
&\cdots \times \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]\Bigg)\Bigg\} \\
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]^T \times \cdots \\
&\cdots\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]\Bigg)\Bigg\} \\
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg(\boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]^T \times \cdots \\
&\cdots\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\boldsymbol{\beta} - \boldsymbol{\beta}^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right] + \cdots \\
&\cdots + \left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]^T\left[\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right]\Bigg)\Bigg\} \\
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg[\left(\boldsymbol{\beta} - \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right)^T\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right) \times \cdots \\
&\cdots \times \left(\boldsymbol{\beta} - \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right)\Bigg]\Bigg\} \\
&\propto \exp\Bigg\{-\frac{1}{2}\Bigg[\left(\boldsymbol{\beta} - \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right)^T\left(\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\right)^{-1} \times \cdots \\
&\cdots \times \left(\boldsymbol{\beta} - \left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}\right)\Bigg]\Bigg\}
\end{aligned}
$$

Which was the kernel to a multivariate normal distribution with a mean $\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y}$ and variance $\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}$ as listed in (D.6).

$$\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\vartheta}_{-\beta} \propto MVN\left(\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{y},\left(\boldsymbol{X}^T\boldsymbol{\Sigma}_{GP}^{-1}\boldsymbol{X}\right)^{-1}\right) \tag{D.6}$$

D.1.3 Derivation of the conditional posterior of $l_m^2|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}$ for the Gaussian process regression

Like with the conditional posterior of s^2, two possible conditional priors were used in deriving the conditional posterior of l_m^2. First, the inverse gamma conditional prior was address and the posterior was listed as D.7. Recall $\boldsymbol{\Sigma}_{GP} = s^2\boldsymbol{C} + s^2\nu^2 I$

$$\begin{aligned}
l_m^2 &\sim \mathcal{IG}\left(n_{l_m^2}, d_{l_m^2}\right)\\
p(l_m^2|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}) &= \frac{p(\boldsymbol{y}|\boldsymbol{X},\vartheta)p(l_m^2|\vartheta_{-l_m^2})}{p(\boldsymbol{y}|\boldsymbol{X},\vartheta_{-l_m^2}}\\
p(\log\left(l_m^2\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}) &\propto p(\boldsymbol{y}|\boldsymbol{X},\vartheta)p(l^2)\times l_m^2\\
p(\log\left(l_m^2\right)|\boldsymbol{y},\boldsymbol{X},\vartheta_{-l_m^2}) &\propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)^T\left(\boldsymbol{\Sigma}_{GP}\right)^{-1}\cdots\right. \qquad \text{(D.7)}\\
&\quad \left.\cdots\times\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\right)\right\}\left(l_m^2\right)^{-n_{l_m^2}-1}\exp\left\{-\frac{d_{l_m^2}}{l_m^2}\right\}\times l_m^2
\end{aligned}$$

The conditional posterior for the case when the conditional prior was the truncated inverse gamma was proportional to D.7 over the support of $\log\left(l_m^2\right)$. Therefore, the same proof can be followed with the addition of the indicator function for $-\infty < \log\left(l_m^2\right) \leq \log(u)$ where u was the upper bound. That result was listed in section 4.2.4.

D.1.4 Derivation of the conditional posterior of $\nu^2|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}$ for the Gaussian process regression

Again, two conditional priors were considered in the derivation of the conditional posterior of ν^2. The conditional posterior of ν^2 was derived when the conditional prior was an inverse gamma. The result was listed as D.8 and $s^2\boldsymbol{C} + s^2\nu^2 I$ was denoted as $\boldsymbol{\Sigma}_{GP}$.

$$\begin{aligned}
\nu^2 &\sim \mathcal{IG}\left(n_{\nu^2}, d_{\nu^2}\right) \\
p\left(\nu^2|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}\right) &= \frac{p(\boldsymbol{y}|\boldsymbol{X}, \vartheta)p\left(\nu^2|\boldsymbol{X}, \vartheta_{-\nu^2}\right)}{p\left(\boldsymbol{y}|\boldsymbol{X}, \vartheta_{-\nu^2}\right)} \\
p\left(\log(\nu^2)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}\right) &\propto p(\boldsymbol{y}|\boldsymbol{X}, \vartheta)p(\nu^2) \times \nu^2 \\
p\left(\log(\nu^2)|\boldsymbol{y}, \boldsymbol{X}, \vartheta_{-\nu^2}\right) &\propto |\boldsymbol{\Sigma}_{GP}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right)^T\left(\boldsymbol{\Sigma}_{GP}\right)^{-1}\cdots\right. \qquad \text{(D.8)} \\
&\left.\cdots \times \left(\boldsymbol{y} - \boldsymbol{X\beta}\right)\right\}\left(\nu^2\right)^{-n_{\nu^2}-1}\exp\left\{-\frac{d_{\nu^2}}{\nu^2}\right\} \times \nu^2
\end{aligned}$$

D.1.5 Derivation of m_{y_*} and Σ_{y_*} for the Gaussian process regression

The derivation of m_{y_*} and Σ_{y_*} was quite lengthy. The derivation was done for the model $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\varepsilon}$ for brevity, but the proof for $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\varepsilon}$ follows the exact form of the latter. First, the posterior distribution of the function $\boldsymbol{f}(\boldsymbol{x})$ was computed.

D.1.5.1 Deriving $\boldsymbol{f}(\boldsymbol{X})|\boldsymbol{y}, \boldsymbol{X}$

$$\boldsymbol{f}(\boldsymbol{X})|\boldsymbol{X}, \vartheta \sim MVN\left(\boldsymbol{0}, s^2C(\boldsymbol{x}, \boldsymbol{x})\right)$$

$$\begin{aligned}
\boldsymbol{y}|\boldsymbol{f(X)},\boldsymbol{X},\vartheta &\sim MVN\left(\boldsymbol{f(x)}, I(s^2\nu^2)\right)\\
p(\boldsymbol{f(X)}|\boldsymbol{y},\boldsymbol{X},\vartheta) &= \frac{p(\boldsymbol{y}|\boldsymbol{f(X)},\boldsymbol{X})p(\boldsymbol{f(X)}|\boldsymbol{X},\vartheta)}{p(\boldsymbol{y}|\boldsymbol{X},\vartheta)}\\
p(\boldsymbol{f(X)}|\boldsymbol{y},\boldsymbol{X},\vartheta) &\propto p(\boldsymbol{y}|\boldsymbol{f(X)},\boldsymbol{X},\vartheta)\pi(\boldsymbol{f(X)}|\boldsymbol{X},\vartheta) \qquad \text{(D.9)}
\end{aligned}$$

For readability, the left from (D.9) was omitted

$$\begin{aligned}
&\propto \exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}-\boldsymbol{f(X)})^T(I(s^2\nu^2))^{-1}(\boldsymbol{y}-\boldsymbol{f(X)}) + (\boldsymbol{f(X)})^T(s^2C)^{-1}(\boldsymbol{f(X)})\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}-\boldsymbol{f(X)})^T(I(s^2\nu^2))^{-1}(\boldsymbol{y}-\boldsymbol{f(X)}) + (\boldsymbol{f(X)})^T(s^2C)^{-1}(\boldsymbol{f(X)})\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(I(s^2C))^{-1}(s^2C)(I(s^2\nu^2))^{-1} + \cdots\right.\right.\\
&\cdots + (s^2C)^{-1}I\sigma_\varepsilon^2(I(s^2\nu^2))^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{y} + \cdots\\
&\left.\left.\cdots\ -\boldsymbol{f(X)}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{y} - \boldsymbol{y}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{f(X)}\right]\right\}
\end{aligned}$$

For brevity, $(s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1} + (s^2C)^{-1}I(s^2\nu^2)(I(s^2\nu^2))^{-1}$ was simplified.

$$\begin{aligned}
(s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1} + (s^2C)^{-1}I(s^2\nu^2)(I(s^2\nu^2))^{-1} &= \cdots\\
\cdots (s^2C)^{-1}((s^2C) + I(s^2\nu^2))(I(s^2\nu^2))^{-1}&\\
\left[(s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1} + (s^2C)^{-1}I(s^2\nu^2)(I(s^2\nu^2))^{-1}\right]^{-1} &= \cdots\\
\cdots I(s^2\nu^2)((s^2C) + I(s^2\nu^2))^{-1}(s^2C)&\\
\left[(s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1} + (s^2C)^{-1}I(s^2\nu^2)(I(s^2\nu^2))^{-1}\right]^{-1} &= \cdots\\
\cdots (s^2\nu^2)((s^2C) + I(s^2\nu^2))^{-1}(s^2C)&
\end{aligned}$$

Let $(s^2\nu^2)((s^2C) + I(s^2\nu^2))^{-1}(s^2C)$ be denoted as Σ_f. Before, returning to

$p(\boldsymbol{f(X)}|\boldsymbol{y}, \boldsymbol{X}, \vartheta)$ consider $\Sigma_f^{-1} s^2 C$.

$$
\begin{aligned}
\Sigma_f^{-1} s^2 C &= \left[(s^2\nu^2)((s^2C) + I(s^2\nu^2))^{-1}(s^2C)\right]^{-1}(s^2C) \\
\Sigma_f^{-1} s^2 C &= (s^2C)^{-1}((s^2C) + I(s^2\nu^2)^2)(s^2\nu^2)^{-1}(s^2C) \\
\Sigma_f^{-1}(s^2C) &= (s^2C)^{-1}(s^2C)(s^2\nu^2)^{-1}(s^2C) + (s^2C)^{-1}I(s^2\nu^2)(s^2\nu^2)^{-1}(s^2C) \\
\Sigma_f^{-1}(s^2C) &= (s^2C)(s^2C)^{-1}(s^2C)(s^2\nu^2)^{-1} + (s^2C)(s^2C)^{-1}I(s^2\nu^2)(s^2\nu^2)^{-1} \\
\Sigma_f^{-1}(s^2C) &= (s^2C)(s^2C)^{-1}((s^2C) + I(s^2\nu^2))(s^2\nu^2)^{-1} \\
\Sigma_f^{-1}(s^2C) &= (s^2C)\Sigma_f^{-1}
\end{aligned}
$$

Therefore, s^2C and Σ_f^{-1} commute. This and the notation was Σ_f substituted into $p(\boldsymbol{f(X)}|\boldsymbol{y}, \boldsymbol{X})$.

$$
\begin{aligned}
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(I(s^2C))^{-1}(s^2C)(I(s^2\nu^2))^{-1} + \cdots\right.\right. \\
&\cdots + (s^2C)^{-1}I\sigma_\varepsilon^2(I(s^2\nu^2))^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{y} + \cdots \\
&\cdots \left.\left. -\boldsymbol{f(X)}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{y} - \boldsymbol{y}^T((s^2C)^{-1}(s^2C)(I(s^2\nu^2))^{-1})\boldsymbol{f(X)}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(\Sigma_f^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T((s^2C)(s^2C)^{-1}(I(s^2\nu^2))^{-1})\boldsymbol{y} + \cdots\right.\right. \\
&\cdots \left.\left. -\boldsymbol{y}^T((I(s^2\nu^2))^{-1}(s^2C)(s^2C)^{-1})\boldsymbol{f(x)}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(\Sigma_f^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T((s^2C)\Sigma_f^{-1}\Sigma_f(s^2C)^{-1}(I(s^2\nu^2))^{-1})\boldsymbol{y}\cdots\right.\right. \\
&\cdots \left.\left. - \boldsymbol{y}^T((I(s^2\nu^2))^{-1}\Sigma_f\Sigma_f^{-1}(s^2C)(s^2C)^{-1})\boldsymbol{f(x)}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(\Sigma_f^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T(\Sigma_f^{-1}(s^2C)\Sigma_f(s^2C)^{-1}(I(s^2\nu^2))^{-1})\boldsymbol{y}\cdots\right.\right. \\
&\cdots \left.\left. -\boldsymbol{y}^T((I(s^2\nu^2))^{-1}\Sigma_f\Sigma_f^{-1}(s^2C)(s^2C)^{-1})\boldsymbol{f(x)}\right]\right\}
\end{aligned}
$$

Then, $\Sigma_f^{-1}(s^2C)\Sigma_f(s^2C)^{-1}(Is^2\nu^2)^{-1}$ and $(Is^2\nu^2)^{-1}\Sigma_f^{-1}\Sigma_f(s^2C)(s^2C)^{-1}$ were simplified.

$$
\begin{aligned}
\Sigma_f^{-1}(s^2C)\Sigma_f(s^2C)^{-1}(Is^2\nu^2)^{-1} &= \Sigma_f^{-1}(s^2C)s^2\nu^2((s^2C)+s^2\nu^2I)^{-1}\frac{(s^2C)(s^2C)^{-1}}{(s^2\nu^2)} \\
&= \Sigma_f^{-1}(s^2C)\frac{s^2\nu^2}{(s^2\nu^2)}((s^2C)+s^2\nu^2I)^{-1}(s^2C)(s^2C)^{-1} \\
&= \Sigma_f^{-1}(s^2C)((s^2C)+s^2\nu^2I)^{-1} \\
(Is^2\nu^2)^{-1}\Sigma_f\Sigma_f^{-1}(s^2C)(s^2C)^{-1} &= (Is^2\nu^2)^{-1}\Sigma_f\Sigma_f^{-1}I \\
&= (Is^2\nu^2)^{-1}s^2\nu^2((s^2C)+s^2\nu^2I)^{-1}(s^2C)\Sigma_f^{-1} \\
&= ((s^2C)+s^2\nu^2I)^{-1}(s^2C)\Sigma_f^{-1} \\
&= \left[((s^2C)+s^2\nu^2I)^{-1}\right]^T(s^2C)^T\Sigma_f^{-1}
\end{aligned}
$$

Then, $\Sigma_f^{-1}(s^2C)((s^2C)+s^2\nu^2I)^{-1}$ and $[((s^2C)+s^2\nu^2I)^{-1}]^T(s^2C)^T\Sigma_f^{-1}$ were substituted into the original.

$$
\begin{aligned}
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(\Sigma_f^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T\Sigma_f^{-1}\left((s^2C)((s^2C)+s^2\nu^2I)^{-1}\right)\boldsymbol{y}\cdots\right.\right. \\
&\left.\left.\cdots -\boldsymbol{y}^T\left(\left[((s^2C)+s^2\nu^2I)^{-1}\right]^T(s^2C)^T\right)\Sigma_f^{-1}\ \boldsymbol{f(x)}\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{f(X)}^T(\Sigma_f^{-1})\boldsymbol{f(X)} - \boldsymbol{f(X)}^T\Sigma_f^{-1}\left((s^2C)((s^2C)+s^2\nu^2I)^{-1}\right)\boldsymbol{y}\cdots\right.\right. \\
&\left.\left.\cdots\left[\left((s^2C)((s^2C)+s^2\nu^2I)^{-1}\right)\boldsymbol{y}\right]^T\Sigma_f^{-1}\boldsymbol{f(x)}\right]\right\}
\end{aligned}
$$

This result was proportional to a normal distribution with mean $(s^2C)((s^2C)+s^2\nu^2I)^{-1}$ and variance Σ_f^{-1}

$$
\begin{aligned}
\boldsymbol{f(X)}|\boldsymbol{y},\boldsymbol{X},\vartheta &\sim MVN\left(m_f,\Sigma_f\right) \\
m_f &= (s^2C)((s^2C)+s^2\nu^2I)^{-1} \\
\Sigma_f &= (s^2\nu^2)((s^2C)+I(s^2\nu^2))^{-1}(s^2C)
\end{aligned}
\tag{D.10}
$$

D.1.5.2 Deriving $\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{y}, \boldsymbol{X}$

Numerically, simulating $\boldsymbol{f(X_*)f(X)}$,$\boldsymbol{y}$,$\boldsymbol{X}$ was difficult because there was no $s^2\nu^2 I$ term in the matrices that needed to be inverted. Therefore, this intermediate step was only used to simplify the derivation. This term should not be sampled given $\boldsymbol{f(X)}$ and in general these intermediate steps were not placed in this thesis to be part of sampler. However, before we can derive the desired $\boldsymbol{y}_*|\boldsymbol{y}, \boldsymbol{X}_*, \boldsymbol{X}$ for prediction, it is necessary to derive $\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_*, \boldsymbol{X}$ first.

$$
\begin{aligned}
p(\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_*) &= \frac{p(\boldsymbol{f(X_*)}, \boldsymbol{f(X)}|\boldsymbol{X}_*, \boldsymbol{X})}{p(\boldsymbol{f(X)}|\boldsymbol{X}_*, \boldsymbol{X})} \\
p(\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_*) &\propto p(\boldsymbol{f(X_*)}, \boldsymbol{f(X)}|\boldsymbol{X}_*, \boldsymbol{X}) \\
\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_* &\propto MVN\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}(s^2C) & (s^2C_*)\\ (s^2C_*)^T & (s^2C_{**})\end{bmatrix}\right)
\end{aligned}
$$

Using the notation $K = s^C$, $K_* = s^2C_*$, and $K_** = s^2C_**$, the covariance matrix was inverted using the block form as above.

$$
\begin{aligned}
\begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}^{-1} &= \begin{bmatrix} A & -K^{-1}K_*\Sigma_{f*|f}^{-1} \\ -\Sigma_{f*|f}^{-1}K_*^T K^{-1} & \Sigma_{f*|f}^{-1} \end{bmatrix} \\
A &= K^{-1} - K^{-1}K_*\left(K_{**} - K_*^T K^{-1}K_*\right)^{-1}K_*^T K^{-1} \\
\Sigma_{f*|f}^{-1} &= \left(K_{**} - K_*^T K^{-1}K_*\right)^{-1}
\end{aligned}
\tag{D.11}
$$

From this, $p(\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_*, \boldsymbol{X})$ was derived from $p(\boldsymbol{f(X_*)}, \boldsymbol{f(X)}|\boldsymbol{X}_*, \boldsymbol{X})$.

$$\boldsymbol{f}(\boldsymbol{X}_*)|\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{X}_*, \boldsymbol{X} \sim MVN\left(\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix}, \begin{bmatrix}K & K_*\\ K_*^T & K_{**}\end{bmatrix}\right)$$

$$\propto \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{f}(\boldsymbol{X})\\ \boldsymbol{f}(\boldsymbol{X}_*)\end{bmatrix}^T \begin{bmatrix}K & K_*\\ K_*^T & K_{**}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{f}(\boldsymbol{X})\\ \boldsymbol{f}(\boldsymbol{X}_*)\end{bmatrix}\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{f}(\boldsymbol{X})\\ \boldsymbol{f}(\boldsymbol{X}_*)\end{bmatrix}^T \begin{bmatrix}A & -K^{-1}K_*\Sigma_{f*|f}^{-1}\\ -\Sigma_{f*|f}^{-1}K_*^TK^{-1} & \Sigma_{f*|f}^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{f}(\boldsymbol{X})\\ \boldsymbol{f}(\boldsymbol{X}_*)\end{bmatrix}\right\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X})^TA + \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}K_*^TK^{-1} + \boldsymbol{f}(\boldsymbol{X})^TK^{-1}K_*\Sigma_{f*|f}^{-1} \cdots$$

$$\cdots + \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}\Big]\begin{bmatrix}\boldsymbol{f}(\boldsymbol{X})\\ \boldsymbol{f}(\boldsymbol{X}_*)\end{bmatrix}\Bigg\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X})^TA\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X}) + \cdots$$

$$\cdots -\boldsymbol{f}(\boldsymbol{X})^TK^{-1}K_*\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*) + \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*)\Big]\Big\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*) - \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X}) + \cdots$$

$$\cdots -\boldsymbol{f}(\boldsymbol{X})^TK^{-1}K_*\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*)\Big]\Big\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*) - \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X}) + \cdots$$

$$\cdots -\boldsymbol{f}(\boldsymbol{X})^TK^{-1}K_*\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*)\Big]\Big\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*) - \boldsymbol{f}(\boldsymbol{X}_*)^T\Sigma_{f*|f}^{-1}K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X}) + \cdots .$$

$$\cdots - \left(K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X})\right)^T\Sigma_{f*|f}^{-1}\boldsymbol{f}(\boldsymbol{X}_*)\Big]\Big\}$$

$$\propto \exp\Big\{-\frac{1}{2}\Big[\boldsymbol{f}(\boldsymbol{X}_*) - K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X})\Big]^T\Sigma_{f*|f}^{-1}\Big[\boldsymbol{f}(\boldsymbol{X}_*) - K_*^TK^{-1}\boldsymbol{f}(\boldsymbol{X})\Big]\Big\}$$

The resulting was proportional to the multivariate normal. The full distribution was defined in D.12.

$$\begin{aligned} & p(\boldsymbol{f(X_*)}|\boldsymbol{f(X)}, \boldsymbol{X}_*, \boldsymbol{X}) \propto MVN_{f(X_*)}\left(\boldsymbol{m}_{f*|f}, \Sigma_{f*|f}\right) \qquad \text{(D.12)} \\ & \Sigma_{f*|f} = K_{**} - K_*^T K^{-1} K_* \\ & \boldsymbol{m}_{f*|f} = K_*^T K^{-1} \boldsymbol{f(X)} \end{aligned}$$

D.1.5.3 Deriving $\boldsymbol{f(X_*)}|\boldsymbol{X}_*, \boldsymbol{y}, \boldsymbol{X}$ and $\boldsymbol{y_*}|\boldsymbol{X}_*, \boldsymbol{y}, \boldsymbol{X}$

As stated above, the true goal is to generate the posteriors for $\boldsymbol{y_*}$. First, the posteriors for $\boldsymbol{f X_*}$ was derived. Since $\boldsymbol{f(X_*)}|\boldsymbol{X}_*, \boldsymbol{f(X)}, \boldsymbol{X}$ was conditional independent of $\boldsymbol{y}$, then $p\left(\boldsymbol{f(X_*)}\,|\boldsymbol{X}_*, \boldsymbol{f(X)}, \boldsymbol{X}\right) = p(\boldsymbol{f(X_*)}|\boldsymbol{X}_*, \boldsymbol{f(X)}, \boldsymbol{y}, \boldsymbol{X})$. Therefore, one can use a combination of the distributions from D.10 and D.12 in conjunction with the laws of total expectation and covariance. Then the law of total expectation was used to find the mean.

$$\begin{aligned} \mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*,y,X}}[\boldsymbol{f(X_*)}] &= \mathbb{E}_{\boldsymbol{f(X)}|fX_*,\boldsymbol{y},\boldsymbol{X}}\left\{\mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{f(X)},\boldsymbol{X_*,y,X}}\left[\boldsymbol{f(X_*)}\right]\right\} \\ \mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*,y,X}}[\boldsymbol{f(X_*)}] &= \mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*,y,X}}\left\{K_*^T K^{-1} \boldsymbol{f(X)}\right\} \\ \mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*,y,X}}[\boldsymbol{f(X_*)}] &= K_*^T K^{-1} \mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*,y,X}}\left\{\boldsymbol{f(X)}\right\} \\ \mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*,y,X}}[\boldsymbol{f(X_*)}] &= K_*^T K^{-1} K(K + (s^2\nu^2 I)^{-1}\boldsymbol{y} \\ \mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*,y,X}}[\boldsymbol{f(X_*)}] &= K_*^T (K + s^2\nu^2 I)^{-1}\boldsymbol{y} \\ m_{f*} &= K_*^T (K + s^2\nu^2 I)^{-1}\boldsymbol{y} \qquad \text{(D.13)} \end{aligned}$$

The law of total covariance was used to compute the covariance matrix. However, it is helpful to write Σ_f in a different from.

$$\Sigma_f = s^2\nu^2(K + s^2\nu^2 I)^{-1} K$$

$$\Sigma_f = \left[K^{-1} + \left(s^2\nu^2 I\right)^{-1}\right]^{-1}$$
$$\Sigma_f = \left[K^{-1} + s^2\nu^2 I \left(s^2\nu^2 I\right)^{-1} \left(s^2\nu^2 I\right)^{-1}\right]^{-1}$$

Then the Woodburry matrix identity which states $(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)V^{-1}A^{-1}$ was applied.

$$\Sigma_f = K - K\left(s^2\nu^2 I\right)\left[s^2\nu^2 I + \left(s^2\nu^2 I\right)^{-1} K s^2\nu^2 I\right]^{-1}\left(s^2\nu^2 I\right)^{-1} K$$
$$\Sigma_f = K - K\left[s^2\nu^2 I + K\right]^{-1} K$$

The law of total covariances applied to this problem here was listed as (D.14), (D.15), and (D.16).

$$Cov_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}] = \cdots \tag{D.14}$$
$$\cdots \mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{Cov_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} + \cdots \tag{D.15}$$
$$\cdots + Cov_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{\mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} \tag{D.16}$$

For brevity, (D.14) was denoted as $Cov[\boldsymbol{f(X_*)}]$ and (D.15) was dealt with separately first.

$$\mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{Cov_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} = \mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{K_{**} - K_*^T K^{-1} K_*\right\}$$
$$\mathbb{E}_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{Cov_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} = K_{**} - K_*^T K^{-1} K_*$$

Then (D.16) was considered.

$$Cov_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{\mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} = Cov_{\boldsymbol{f(X)}|\boldsymbol{X_*},\boldsymbol{y},\boldsymbol{X}}\left\{K_*^T K^{-1}\boldsymbol{f(X)}\right\}$$

$$Cov_{\boldsymbol{f(X)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}}\left\{\mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} = K_*^T K^{-1} Cov_{\boldsymbol{f(X)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}}\left[\boldsymbol{f(X)}\right] K^{-1} K_*$$

$$Cov_{\boldsymbol{f(X)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}}\left\{\mathbb{E}_{\boldsymbol{f(X_*)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}}[\boldsymbol{f(X_*)}]\right\} = \cdots$$

$$\cdots K_*^T K^{-1}\left[K - K\left(s^2\nu^2 I + K\right)^{-1} K\right] K^{-1} K_*$$

The simplified forms of (D.15 and (D.16) were used to compute (D.14).

$$Cov[\boldsymbol{f(X_*)}] = K_{**} - K_*^T K^{-1} K_* + K_*^T K^{-1}\left[K - K\left(s^2\nu^2 I + K\right)^{-1} K\right] K^{-1} K_*$$

$$Cov[\boldsymbol{f(X_*)}] = K_{**} - K_*^T K^{-1} K_* + K_*^T K^{-1} K K^{-1} K_* + \cdots$$

$$\cdots - K_*^T K^{-1} K\left(s^2\nu^2 I + K\right)^{-1} K K^{-1} K_*$$

$$Cov[\boldsymbol{f(X_*)}] = K_{**} - K_*^T K^{-1} K_* + K_*^T K^{-1} K_* - K_*^T\left(s^2\nu^2 I + K\right)^{-1} K_*$$

$$Cov[\boldsymbol{f(X_*)}] = K_{**} - K_*^T\left(s^2\nu^2 I + K\right)^{-1} K_*$$

Also, since $p(\boldsymbol{f(X_*)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X})$, was a convolution of two multivariate normals and therefore also normal, the covariance and expectation was sufficient to describe the posterior of $\boldsymbol{f(X)}_*$.

$$\begin{aligned}\boldsymbol{f(X_*)}|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X} &\sim MVN\left(m_{f_*},\Sigma_{f_*}\right) \\ m_{f_*} &= K_*^T(K + s^2\mu^2 I)^{-1}\boldsymbol{y} \\ \Sigma_{f_*} &= K_{**} - K_*^T\left(s^2\nu^2 I + K\right)^{-1} K_*\end{aligned} \tag{D.17}$$

From this, deriving $\boldsymbol{y}_*|\boldsymbol{X}_*,\boldsymbol{y},\boldsymbol{X}$ was straight forward.

$$\boldsymbol{y}_* = \boldsymbol{f(X_*)} + \boldsymbol{\varepsilon}_*$$

$$\mathbb{E}[\boldsymbol{y}_*] = \mathbb{E}[\boldsymbol{f(X_*)} + \boldsymbol{\varepsilon}_*]$$

$$\mathbb{E}[\boldsymbol{y}_*] = \mathbb{E}[\boldsymbol{f(X_*)}] + \mathbb{E}[\boldsymbol{\varepsilon}_*]$$

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{y}_{*}] &= m_{f*} + 0 \\
m_{\boldsymbol{y}_{*}} &= K_{*}^{T}(K + s^{2}\nu^{2}I)^{-1}\boldsymbol{y} \qquad \text{(D.18)} \\
Cov(\boldsymbol{y}_{*}) &= Cov(\boldsymbol{f}(\boldsymbol{X}_{*}) + \boldsymbol{\varepsilon}_{*}) \\
Cov(\boldsymbol{y}_{*}) &= Cov(\boldsymbol{f}(\boldsymbol{X}_{*})) + Cov(\boldsymbol{\varepsilon}_{*}) \\
Cov(\boldsymbol{y}_{*}) &= K_{**} - K_{*}^{T}\left(s^{2}\nu^{2}I + K\right)^{-1}K_{*} + s^{2}\nu^{2}I \\
\Sigma_{\boldsymbol{y}_{*}} &= K_{**} - K_{*}^{T}\left(s^{2}\nu^{2}I + K\right)^{-1}K_{*} + s^{2}\nu^{2}I \qquad \text{(D.19)}
\end{aligned}
$$

The distribution of $\boldsymbol{y}_{*}|\boldsymbol{X}_{*}, \boldsymbol{y}, \boldsymbol{X}$ was also normal, making (D.18) and (D.19) a sufficient description. In addition, the posterior distribution in D.20 was listed in the terms defined with the original problem from section 4.2.4.

$$
\begin{aligned}
\boldsymbol{y}_{*}|\boldsymbol{X}_{*}, \boldsymbol{y}, \boldsymbol{X} &\sim MVN\left(m_{\boldsymbol{y}_{*}}, \Sigma_{\boldsymbol{y}_{*}}\right) \qquad \text{(D.20)} \\
m_{\boldsymbol{y}_{*}} &= s^{2}C_{*}^{T}(s^{2}C + s^{2}\nu^{2}I)^{-1}\left(\boldsymbol{y} - \boldsymbol{X\beta}\right) \\
\Sigma_{\boldsymbol{y}_{*}} &= s^{2}C_{**} - s^{2}C_{*}^{T}\left(s^{2}\nu^{2}I + s^{2}C\right)^{-1}s^{2}C_{*} + s^{2}\nu^{2}I
\end{aligned}
$$

D.1.5.4 Derivation of $\boldsymbol{x}_{*m}|\boldsymbol{y}, \boldsymbol{X}, y_{*}, \vartheta$ for the Gaussian process regression

The derivation for the general case with one response variable was addressed first. For this, the conditional prior suggested for the general case introduced in section 4.2.4 was used.

$$
x_{*m} \sim \mathcal{TN}\left(\frac{\sum_{i=1}^{N} x_{im}}{N}, \tau_{*}^{2}, [a, b]\right)
$$

$$
p(x_{*m}|\boldsymbol{X}, \boldsymbol{y}, y_{*}, (\boldsymbol{x}_{*})_{-x_{*m}}, \vartheta) = \frac{p(y_{*}|\boldsymbol{x}_{*}, \boldsymbol{X}, \boldsymbol{y}, \vartheta)p(x_{*m}|\boldsymbol{X}, \boldsymbol{y}, (\boldsymbol{x}_{*})_{-x_{*m}}, \vartheta)}{p(y_{*}|\boldsymbol{X}, \boldsymbol{y}, (\boldsymbol{x}_{*})_{-x_{*m}}, \vartheta)}
$$

$$p(x_{*m}|\boldsymbol{X}, \boldsymbol{y}, y_*, (\boldsymbol{x}_*)_{-x_{*m}}, \vartheta) \propto p(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}, \vartheta)p(x_{*m})$$

$$p\left(x_{*m} \mid \boldsymbol{y}, \boldsymbol{X}, y_*, (\boldsymbol{x}_*)_{-x_{*m}}, \vartheta\right) \propto |\boldsymbol{\Sigma}_{y_*}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y_* - m_{y_*})^T \times \cdots\right.$$

$$\left.\cdots \times (\boldsymbol{\Sigma}_{y_*})^{-1}(y_* - m_{y_*})\right\} \times \frac{1}{\tau_*}\exp\left\{-\frac{\left(x_{*m} - \frac{1}{N}\left(\sum_{i=1}^{N} x_{im}\right)\right)^2}{2\tau_*^2}\right\} \times \cdots$$

$$\cdots \times I[a \leq q_{ij} \leq b]$$

The case of the multivariate response when the response variables were independent was very similar. Since the multivariate response when the response variables were independent was the what was applied for this research, notation specific to the research was used.

$$q_{*ij} \sim \mathcal{TN}\left(\hat{q}_{ij}, \tau_*^2, [a,b]\right)$$

$$p\left(q_{*ij}|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right) = \frac{p\left(\hat{\boldsymbol{\theta}}_{*q}|\boldsymbol{\theta}_{*q}, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q, \vartheta\right) p\left(q_{*ij}|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right)}{p\left(\hat{\boldsymbol{\theta}}_{*q}|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right)}$$

$$p\left(\left(q_{*ij}\right)_p|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right) \propto p\left(\hat{\boldsymbol{\theta}}_{*q}|\boldsymbol{\theta}_{*q}, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q, \vartheta\right) p\left(q_{*ij}\right)$$

$$p\left(\left(q_{*ij}\right)_p|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right) \propto \left[\prod_{\forall \hat{q}_{*ij} \in \hat{\boldsymbol{\theta}}_{*q}} p\left(\hat{q}_{*ij}|\boldsymbol{\theta}_{*q}, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_q, \vartheta\right)\right] p\left(q_{*ij}\right)$$

$$p\left(\left(q_{*ij}\right)_p|\hat{\boldsymbol{\theta}}_q, \boldsymbol{\theta}_q, \hat{\boldsymbol{\theta}}_{*q}, \boldsymbol{\theta}_{*q,-q_{*ij}}, \vartheta\right) \propto I[a \leq q_{*ij} \leq b] \times \cdots$$

$$\cdots \times \left[\prod_{\forall \hat{q}_{*ij} \in \hat{\boldsymbol{\theta}}_{*q}} \frac{1}{\sigma_{q_{*ij}}} \exp\left\{-\frac{\left(\hat{q}_{*ij} - m_{\hat{q}_{*ij}}\right)^2}{2\sigma^2_{\hat{q}_{*ij}}}\right\}\right] \times \frac{1}{\tau_*}\exp\left\{-\frac{\left(q_{*ij} - \hat{q}_{*ij}\right)^2}{2\tau_*^2}\right\}$$

$$m_{\hat{q}_{*ij}} = s^2_{q_{ij}} C^T_{*q_{ij}} (s^2_{q_{ij}} C_{q_{ij}} + s^2_{q_{ij}} \nu^2_{q_{ij}} I)^{-1} \left(\hat{\boldsymbol{q}}_{ij} - \boldsymbol{\theta_q} \boldsymbol{\beta_{q_{ij}}}\right)$$

$$\sigma^2_{\hat{q}_{*ij}} = s^2_{q_{ij}} C_{**q_{ij}} - s^2_{q_{ij}} C^T_{*q_{ij}} \left(s^2_{q_{ij}} \nu^2_{q_{ij}} I + s^2_{q_{ij}} C_{q_{ij}}\right)^{-1} s^2_{q_{ij}} C_{*q_{ij}} + s^2_{q_{ij}} \nu^2_{q_{ij}} I$$

D.2 Additional analysis on the bias reduction using $\left(\log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_q\right)\right)$

The dataset used to generate a posterior for $\log\left(\hat{\boldsymbol{\theta}}_{*q}\right)$ had a range of $\log\left(q_{*12}\right) \in [-7.2735, -6.4342]$, $\log\left(q_{*21}\right) \in [-5.1967, -4.3077]$, $\log\left(q_{*23}\right) \in [-7.8118, -6.0627]$ and $\log\left(q_{*32}\right) \in [-7.2223, -5.3542]$. Therefore, given the length scale definition of $\boldsymbol{l}^2_{q_{ij}}$, $\boldsymbol{l}^2_{q_{ij}}$ should be restricted to a small range. However, for the posterior of q_{ij} it was possible that some predictor variables could have little or no correlation with q_{ij}. Given the Gaussian process model used, this make it necessary to include fairly large $\boldsymbol{l}^2_{q_{ij}}$. Therefore, the prior listed as D.21 was used to allow for $\boldsymbol{l}^2_{q_{ij}}$ to act as a weight and a length scale without exploring the entirety of the long and correlated tails and the parameters were chosen to be "slightly informative". Given the ranges, σ^2_ε should also be limited as it would not make sense to have noise bigger than the range. As with $\boldsymbol{l}^2_{q_{ij}}$, both $s^2_{q_{ij}}$ and $\nu^2_{q_{ij}}$ were chosen to allow a fair amount of exploration without allowing exploration into the long correlated tails. The priors for $s^2_{q_{ij}}$ and $\nu^2_{q_{ij}}$ were listed as (D.22) and (D.23) respectively.

$$l^2_{q_{ij},m} \sim \mathcal{IG}\left(1, 1, 50\right) \tag{D.21}$$

$$s^2 \sim \mathcal{IG}\left(1, 1, 100\right) \tag{D.22}$$

$$\nu^2 \sim \mathcal{IG}\left(1, 1, 10\right) \tag{D.23}$$

The histograms of the posterior were placed in D.1.

From D.1, it appears that $l^2_{q_{ij},1}$ and $l^2_{q_{ij},2}$ have shorter length scales where $l^2_{q_{ij},3}$ and $l^2_{q_{ij},4}$ have longer length scales for all q_{ij} in $\boldsymbol{\theta}_q$. For any q_{ij}, $l^2_{q_{ij},m}$ may be slightly shorter than $l^2_{q_{kl},m}$ for all $k, l \neq i, j$ if m was important to q_{ij}, but the effect was smaller than the general length scale of that variable. In most case

The histograms of the simulated $l^2_{q_{ij},m} | \log\left(\hat{\boldsymbol{\theta}}_q\right), \log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_{*q}\right), \vartheta_{\log,-l^2_{q_{ij},m}}$

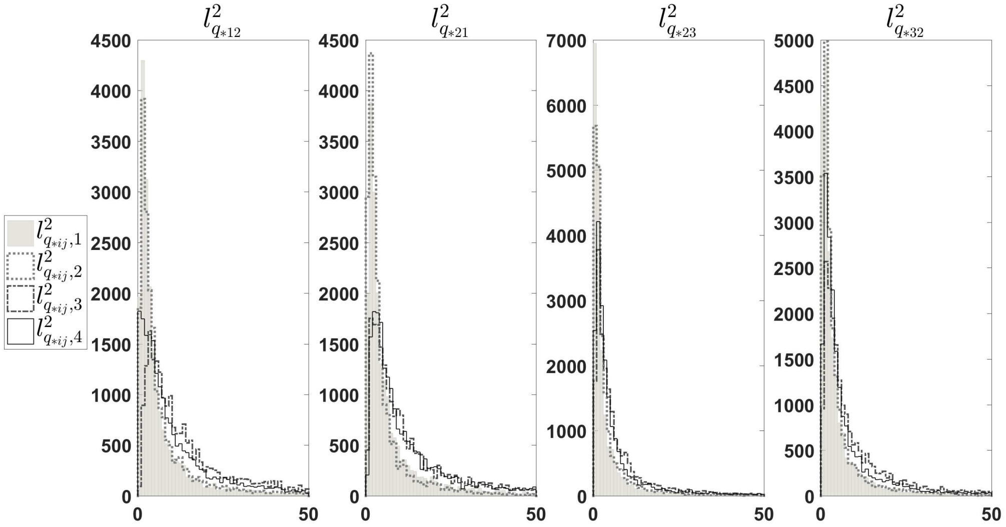


Figure D.1: The histogram of $l^2_{q_{ij},1}$ was filled in where $l^2_{q_{ij},2} - l^2_{q_{ij},4}$ were outlines as described in the key

$l^2_{q_{ij},m}$ seemed to be close to the same for all q_{ij} in $\boldsymbol{\theta}_q$.. Therefore, it appears that most of the weight or importance of the variable was described in $\boldsymbol{\beta}_{q_{ij}}$. Finally, $\sigma^2_{\varepsilon,q_{ij}}$ acted as expected and was place in figure D.2. There was more noise or larger $\sigma^2_{\varepsilon,q_{ij}}$ for q_{23} and q_{32} and less for q_{21} and q_{12}.

The histograms of the simulated $s^2_{q_{ij}} \nu^2_{q_{ij}} | \log\left(\hat{\boldsymbol{\theta}}_q\right), \log\left(\boldsymbol{\theta}_q\right), \log\left(\hat{\boldsymbol{\theta}}_{*q}\right), \vartheta_{\log,-l^2_{q_{ij},m}}$

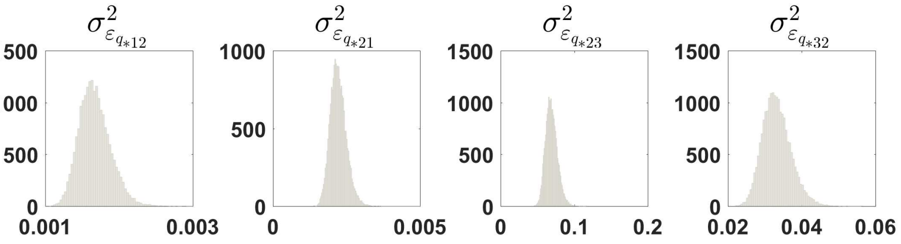


Figure D.2: The histogram of $s^2_{q_{12}} \nu^2_{q_{12}}$ or $\sigma^2_{\varepsilon,q_{ij}}$

Bibliography

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

[2] Mark Akeson, Daniel Branton, John J. Kasianowicz, Eric Brandin, and David W. Deamer. Microsecond time-scale discrimination among polycytidylic acid, polyadenylic acid, and polyuridylic acid as homopolymers or as segments within single {RNA} molecules. *Biophysical Journal*, 77(6):3227 – 3233, 1999.

[3] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, Jan 2003.

[4] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[5] Seico Benner, Roger J A Chen, Noah A Wilson, Robin Abu-Shumays, Nicholas Hurt, Kate R Lieberman, David W Deamer, William B Dunbar, and Mark Akeson. Sequence-specific detection of individual dna polymerase complexes in real time using a nanopore. *Nature nanotechnology*, 2(11):718–724, 2007. 1748-3387.

[6] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[7] O. Bousquet, U. von Luxburg, and G. Rätsch. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011.

[8] A.W. Bush. *Perturbation Methods for Engineers and Scientists*. CRC Press library of engineering mathematics. Taylor & Francis, 1992.

[9] Siddhartha Chib. Calculating posterior distributions and modal estimates in markov mixture models. Journal of Econometrics, 75(1):79–97, 1996.

[10] S. H. Chung, John B. Moore, Lige Xia, L. S. Premkumar, and P. W. Gage. Characterization of single channel currents using digital signal processing techniques based on hidden markov models. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 329(1254):265–285, 1990.

[11] Joseph M. Dahl, Ai H. Mai, Gerald M. Cherf, Nahid N Jetha, Daniel R Garalde, Andre Marziali, Mark Akeson, Hongyun Wang, and Kate R Lieberman. Direct observation of translocation in individual dna polymerase complexes. The Journal of Biological Chemistry, 287:13407–13421, 2012.

[12] John D'Errico. nearestspd for matlab.

[13] Sylvie Doublie, Stanley Tabor, Alexander M. Long, Charles C. Richardson, and Tom Ellenberger. Crystal structure of a bacteriophage t7 dna replication complex at 2.2 a resolution. Nature, 391:251–258, 1998.

[14] N.R. Draper and H. Smith. Applied regression analysis. Number v. 1 in Wiley series in probability and statistics: Texts and references section. Wiley, 1998.

[15] Andrew M. Fraser. Hidden Markov models and dynamical systems. Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, Philadelphia, Pa., 2008.

[16] Sylvia FrÃijhwirth-Schnatter. Data augmentation and dynamic linear models. Journal of Time Series Analysis, 15(2):183–202.

[17] Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. Bayesian Data Analysis, Third Edition (Chapman & Hall/CRC Texts in Statistical Science). Chapman and Hall/CRC, London, third edition, November 2014.

[18] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell., 6(6):721–741, 1984.

[19] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics, 22(4):403 – 434, 1976.

[20] Robert B Gramacy. Bayesian treed Gaussian process models. PhD thesis, University of California, Santa Cruz, 2005.

[21] Robert B. Gramacy and Herbert K. H. Lee. Cases for the nugget in modeling computer experiments. Statistics and Computing, 22(3):713–722, May 2012.

[22] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. Biometrika, 57(1):97–109, 1970.

[23] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. Linear Algebra and its Applications, 103:103 – 118, 1988.

[24] F. Jelinek. Continuous speech recognition by statistical methods. Proceedings of the IEEE, 64(4):532–556, April 1976.

[25] J. L. W. V. Jensen. Sur les fonctions convexes et les inÃľgalitÃľs entre les valeurs moyennes. Acta Math., 30:175–193, 1906.

[26] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.

[27] Marc Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. 63:425–464, 02 2001.

[28] J.P.C. Kleijnen. Design and Analysis of Simulation Experiments. International Series in Operations Research & Management Science. Springer International Publishing, 2015.

[29] Neil D. Lawrence, Magnus Rattray, and Michalis K. Titsias. Efficient sampling for gaussian process inference using control variables. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1681–1688. Curran Associates, Inc., 2009.

[30] Kate R. Lieberman, Gerald M. Cherf, Michael J. Doody, Felix Olasagasti, Yvette Kolodji, and Mark Akeson. Processive replication of single dna molecules in a nanopore catalyzed by phi29 dna polymerase. Journal of the American Chemical Society, 132(50):17961–17972, 2010. PMID: 21121604.

[31] Kate R. Lieberman, Joseph M. Dahl, Ai H. Mai, Mark Akeson, and Hongyun Wang. Dynamics of the translocation step measured in individual dna polymerase complexes. Journal of the American Chemical Society, 134(45):18816–18823, 2012. PMID: 23101437.

[32] Kate R. Lieberman, Joseph M. Dahl, Ai H. Mai, Ashley Cox, Mark Akeson, and Hongyun Wang. Kinetic mechanism of translocation and dntp binding in individual dna polymerase complexes. Journal of the American Chemical Society, 135(24):9149–9155, 2013. PMID: 23705688.

[33] Lawrence A. Loeb. Mutator phenotype may be required for multistage carcinogenesis. Cancer Research, 51(12):3075–3079, 1991.

[34] Iain L. MacDonald. Hidden Markov and other models for discrete-valued time series. Monographs on statistics and applied probability ; 70. Chapman /Hall, London, first edition. edition, 1997.

[35] Georges Matheron. Principles of geostatistics. Economic geology, 58(8):1246–1266, 1963.

[36] M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2):239–245, 1979.

[37] G. McLachlan and T. Krishnan. The EM Algorithm and Extensions. Wiley Series in Probability and Statistics. Wiley, 1996.

[38] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6):1087–1092, 1953.

[39] Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. arXiv preprint physics/9701026, 1997.

[40] Anthony O'Hagan, JM Bernardo, JO Berger, AP Dawid, AFM Smith, et al. Uncertainty analysis and other inference tools for complex computer codes. Bayesian Statistics, 1998.

[41] Raquel Prado and Mike West. Time Series: Modeling, Computation, and Inference. Chapman & Hall/CRC, 1st edition, 2010.

[42] Feng Qin. Restoration of single-channel currents using the segmental k-means method based on hidden markov modeling. Biophysical journal, 86 3:1488–501, 2004.

[43] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, Feb 1989.

[44] L. R. Rabiner, J. G. Wilpon, and B. H. Juang. A segmental k-means training procedure for connected word recognition. AT T Technical Journal, 65(3):21–31, May 1986.

[45] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, 2006.

[46] Christian Robert and George Casella. Introducing Monte Carlo Methods with R. Springer, 2010.

[47] Christian Robert and George Casella. A short history of markov chain monte carlo: Subjective recollections from incomplete data. Statistical Science, 26(1):102–115, 2011.

[48] Christian P. Robert, Gilles Celeux, and Jean Diebolt. Bayesian estimation of hidden markov chains: a stochastic implementation. Statistics & Probability Letter, 16(1):77 – 83, 1993.

[49] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. Statist. Sci., 4(4):409–423, 11 1989.

[50] Margarita Salas, Luis Blanco, JosÃľ M. LÃązaro, and Miguel de Vega. The bacteriophage phi 29 dna polymerase. IUBMB Life, 60(1):82–85, 2008.

[51] T.J. Santner, B.J. Williams, and W.I. Notz. The Design and Analysis of Computer Experiments. Springer Series in Statistics. Springer New York, 2010.

[52] Frühwirth S. Schnatter. Finite mixture and Markov switching models. Springer Verlag, 2006.

[53] Steven L. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. Journal of the American Statistical Association, 97(457):337–351, 2002.

[54] Robert H. Shumway. Time series analysis and its applications : with R examples. Springer texts in statistics. Springer, New York, second [updated] edition. edition, 2006.

[55] F.K. Tai, Run ze Li, and A. Sudjianto. Design And Modeling for Computer Experiments. Chapman and Hall/CRC Computer Science and Data Analysis Series. Chapman & Hall/CRC, 2006.

[56] D.M. Titterington, A.F.M. Smith, and U.E. Makov. Statistical Analysis of Finite Mixture Distributions. Wiley, New York, 1985.

[57] G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. Phys. Rev., 36:823–841, Sep 1930.

[58] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2):260–269, April 1967.

[59] M. West and J. Harrison. Bayesian forecasting and dynamic models. Springer series in statistics. Springer, 1989.

[60] H. Zhang, W. Cao, E. Zakharova, W. Konigsberg, and E. M. De La Cruz. Fluorescence of 2-aminopurine reveals rapid conformational changes in the rb69 dna polymerase-primer/template complexes upon binding and incorporation of matched deoxynucleoside triphosphates. Nucleic Acids Research, 35(18):6052–6062, 2007.