

---

# Modelling Signal Interactions with Application to Financial Time Series

by

Bonny Jain

---

Submitted to the Department of Electrical Engineering and Computer Science  
in partial fulfillment of the requirements for the degree of

Master of Engineering

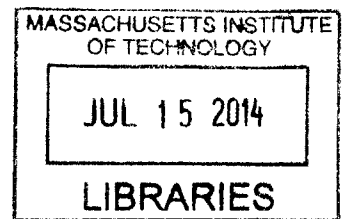
in

Electrical Engineering and Computer Science  
at the Massachusetts Institute of Technology

June 2014

© 2014 Massachusetts Institute of Technology  
All Rights Reserved.

**ARCHIVES**



Signature of Author: Signature redacted

Department of Electrical Engineering and Computer Science


May 23, 2014

Certified by: Signature redacted

John W. Fisher III, Senior Research Scientist of EECS

Thesis Supervisor

Accepted by: Signature redacted

  
Albert R. Meyer, Professor of EECS

Chairman, Masters of Engineering Thesis Committee



---

---

# Modelling Signal Interactions with Application to Financial Time Series

by Bonny Jain

Submitted to the Department of Electrical Engineering and Computer Science  
in partial fulfillment of the requirements for the degree of  
Master of Engineering

## Abstract

In this thesis, we concern ourselves with the problem of reasoning over a set of objects evolving over time that are coupled through interaction structures that are themselves changing over time. We focus on inferring time-varying interaction structures among a set of objects from sequences of noisy time series observations with the caveat that the number of interaction structures is not known a priori. Furthermore, we aim to develop an inference procedure that operates *online*, meaning that it is capable of incorporating observations as they arrive.

We develop an online nonparametric inference algorithm called Online Nonparametric Switching Temporal Interaction Model inference (ONSTIM). ONSTIM is an extension of the work of Dzunic and Fisher [1], who employ a linear Gaussian model with time-varying transition dynamics as the generative graphical model for observed time series. Like Dzunic and Fisher, we employ sampling approaches to perform inference. Instead of presupposing a fixed number of interaction structures, however, we allow for proposal of new interaction structures sampled from a prior distribution as new observations are incorporated into our inference.

We then demonstrate the viability of ONSTIM on synthetic and financial datasets. Synthetic datasets are sampled from a generative model, and financial datasets are constructed from the price data of various US stocks and ETFs.

---

Thesis Supervisor: John W. Fisher III

Title: Senior Research Scientist of Electrical Engineering and Computer Science





---

---

# Acknowledgments

I would like to thank my advisor, Dr. John W. Fisher, for the effort spent in guiding me through my thesis. His ideas, especially those regarding Bayesian nonparametrics, were key to the development of the work detailed in this thesis and also to the development of my own way of thinking.

I would like to thank my senior student collaborator Zoran Dzunic for putting up with me through all the times I barged into his office, and for his willingness to put down whatever he was doing to answer my questions. Zoran effectively taught me all the relevant background material for my thesis, and this work would not have been possible without him.

Finally, I would like to thank all of my group members for providing me with an awesome experience this year. I would like to especially thank Georgios Papachristoudis, who contributed to my education in inference as my TA for 6.438, and Jason Chang, who introduced me to the Fisher group and made it possible for me to run my experiments. I've learned so much from my entire group this year, and I feel very lucky to have had this opportunity.



---

---

# Contents

<b>Abstract</b>	<b>3</b>
<b>Acknowledgments</b>	<b>4</b>
<b>Contents</b>	<b>7</b>
<b>List of Figures</b>	<b>11</b>
<b>List of Algorithms</b>	<b>13</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Thesis Outline . . . . .	16
<b>2 Background</b>	<b>19</b>
2.1 Graphical Models . . . . .	19
2.1.1 Undirected Graphical Models . . . . .	20
2.1.2 Directed Graphical Models . . . . .	20
2.2 Inference . . . . .	21
2.2.1 Belief Propagation . . . . .	21
2.2.2 Sampling . . . . .	23
2.2.3 Conjugate Priors . . . . .	27
2.3 Switching State-Space Interaction Model: Graphical Model . . . . .	31
2.3.1 Dynamic Bayesian Networks . . . . .	31
2.3.2 Linear Gaussian State Space Interaction Model . . . . .	32
2.3.3 Graphical Model . . . . .	34
2.4 Switching State-Space Temporal Interaction Model: Inference . . . . .	39

2.4.1	Sample $X \sim P(X Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ . . . . .	39
2.4.2	Sample $Z \sim P(Z X, \tilde{E}, \tilde{\theta}, \pi)$ . . . . .	42
2.4.3	Sample $\tilde{E}, \tilde{\theta} \sim P(\tilde{E}, \tilde{\theta} Z, X; \beta, \gamma)$ . . . . .	43
2.4.4	Sample $\pi \sim P(\pi Z; \alpha)$ . . . . .	44
2.4.5	Sample $\xi \sim P(\xi X, Y; \delta)$ . . . . .	44
2.4.6	Initialization . . . . .	44
2.5	Summary . . . . .	45
<b>3</b>	<b>Online Nonparametric Switching Temporal Interaction Model</b>	<b>47</b>
3.1	Motivation . . . . .	48
3.2	State Sequence Generative Model . . . . .	49
3.3	Overview of ONSTIM Inference . . . . .	55
3.4	Complexity of Exact Inference . . . . .	59
3.4.1	Intractable Message Passing for $X$ . . . . .	60
3.4.2	Alternative Approaches . . . . .	61
3.5	Batch Initialization of $X$ and $Z$ : Approach 1 . . . . .	62
3.5.1	Initialization of $Z_i \sim P(Z_i Z_{i-1}, Y_i, X_{i-1})$ . . . . .	63
3.5.2	Initialization of $X_i \sim P(X_i X_{i-1}, Z_i, Y_i)$ . . . . .	65
3.6	Batch Initialization of $X$ and $Z$ : Approach 2 . . . . .	66
3.7	Gibbs Sampler . . . . .	68
3.8	Summary . . . . .	69
<b>4</b>	<b>Results</b>	<b>71</b>
4.1	Empirical Model Characterization . . . . .	71
4.1.1	Synthetic Dataset Generation . . . . .	72
4.1.2	Inferred Number of Switching States . . . . .	72
4.1.3	Discussion . . . . .	74
4.2	Experiments with Financial Datasets . . . . .	79
4.2.1	S&P100: 2007-2012 . . . . .	80
4.2.2	S&P100: Flash Crash . . . . .	81
4.2.3	S&P Sector ETFs . . . . .	85
4.3	Summary . . . . .	94
<b>5</b>	<b>Conclusion</b>	<b>95</b>

CONTENTS

9

---

5.1 Drawbacks . . . . .	96
5.2 Further Work . . . . .	96
5.3 Concluding Remarks . . . . .	97

**Bibliography**

**99**



---



---

## List of Figures

2.1	Dynamic Bayesian Network (DBN) . . . . .	32
2.2	Switching State-space Interaction Model (SSIM) . . . . .	36
3.1	Dirichlet Distributions . . . . .	50
3.2	Switching Sequence Generative Model . . . . .	51
3.3	$K$ Markov Chain Model . . . . .	53
3.4	Distribution of $K_t$ . . . . .	54
3.5	$\bar{K}$ -Contour Map . . . . .	55
3.6	Batch Sampling . . . . .	57
4.1	Example Synthetic Dataset . . . . .	73
4.2	$P(\hat{K} K)$ for low $\alpha_{new}, \alpha_{self}$ . . . . .	75
4.3	$P(\hat{K} K)$ for high $\alpha_{new}, \alpha_{self}$ . . . . .	76
4.4	Conditional Bias of $\hat{K}$ for low $\alpha_{new}, \alpha_{self}$ . . . . .	77
4.5	Conditional Bias of $\hat{K}$ for high $\alpha_{new}, \alpha_{self}$ . . . . .	78
4.6	S&P100: New States . . . . .	82
4.7	S&P100: Total States . . . . .	83
4.8	S&P100: SSM . . . . .	84
4.9	Flash Crash: New States . . . . .	85
4.10	Flash Crash: Total States . . . . .	86
4.11	Flash Crash: SSM for low $\alpha_{new}, \alpha_{self}$ . . . . .	87
4.12	Flash Crash: SSM for high $\alpha_{new}, \alpha_{self}$ . . . . .	88
4.13	S&P Sectors: New States . . . . .	90
4.14	S&P Sectors: Total States . . . . .	91

4.15 S&P Sectors: SSM . . . . .	92
4.16 S&P Sectors: Edge Posteriors . . . . .	93



---

---

## List of Algorithms

1	Belief Propagation . . . . .	22
2	Metropolis-Hastings . . . . .	26
3	Gibbs Sampling . . . . .	27
4	SSIM Gibbs Sampler . . . . .	39
5	Sample $X \sim P(X Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ . . . . .	40
6	Sample $Z \sim P(Z X, \tilde{E}, \tilde{\theta}, \pi)$ . . . . .	42
7	ONSTIM . . . . .	59
8	Batch Initialization: Approach 1 . . . . .	63
9	Batch Initialization: Approach 2 . . . . .	68
10	Post-initialization Gibbs sampler . . . . .	69



# Introduction

In fields as diverse as particle physics, molecular biology, and finance, an important problem is determining the relationships among the objects in a system from observations of their behavior. Whether the objects of consideration are subatomic particles interacting via electromagnetic and nuclear forces, genes and proteins interacting through regulatory networks, or financial instruments interacting through market forces, understanding the structure of the interactions among the objects can lend valuable insight into the system as a whole.

Inferring interaction structures can be difficult since there is often no way to *directly* observe the interactions themselves. Instead, we typically have observations of the time-varying trajectories of each object through its state space, such as the position of a particle, the expression level of a protein, or the price of a financial instrument. Interaction structures must then be inferred from these individual trajectories. As a further complication, object trajectory observations are typically noisy, requiring the additional step of inferring the true trajectory from the noisy observations.

Moreover, the interaction structures among a set of objects are not necessarily static, but can instead change over time. For example, suppose three children Alice, Bob, and Charlie are playing tag in a schoolyard, and an observer is tracking their positions but does not know who the chaser is. Suppose Alice is initially the chaser, so the interaction structure is Alice following Bob and Charlie. At some point, Alice successfully tags Bob, causing the interaction structure to switch to Bob following Alice and Charlie. As the observer, inference of such *time-varying* interaction structures from potentially noisy observations adds yet another layer of complexity to the problem we have described thus far.

Siracusa and Fisher [8] modelled the time-varying interaction structures using graph-

ical models, and then developed algorithms to perform inference on these models with sampling techniques. Dzunic and Fisher [1] then extended the graphical models and corresponding inference algorithms to account for noisy and potentially missing observations. An important question that arises when modeling time-varying interaction structures is that of model complexity - how many different interaction structures are sufficient to explain the patterns in the observed data? In both the original work of Siracusa and Fisher and the subsequent work of Dzunic and Fisher, model complexity is user-specified. That is, inference on the model requires prior specification of the number of active interaction structures during the time in which observations are taken. However, the number of different interaction structures is often unknown, and it is then desirable to use inference algorithms that eliminate user-specified model complexity in favor of learning it automatically from the observed data.

In this thesis, we consider the problem of inferring the structure of relationships among a set of covarying time series from a sequence of noisy observations when the number of interaction structures is not known a priori. We take inspiration from literature on Bayesian nonparametrics, a growing field of statistics aimed at increasing flexibility of model parameter specification, which typically implements such flexibility by learning parameters from the data. Furthermore, our approach allows for the incorporation of observations into the inference procedure as they arrive, instead of requiring knowledge of all observations before performing any inference. Such approaches are called online algorithms, and by using such an approach, we make a tradeoff between the speed of inference and the accuracy of the results. We characterize the performance of our approach on synthetic and real datasets, and we discuss its merits and drawbacks for various applications.

## ■ 1.1 Thesis Outline

We develop an online algorithm to perform inference over interaction structures that decides model complexity nonparametrically. We call this algorithm the Online Nonparametric Switching Temporal Interaction Model inference algorithm, which we abbreviate as ONSTIM. In Chapter 2, we discuss background material relevant to the development of ONSTIM. In Chapter 3, we describe a generative model that proposes a mechanism by which new interaction structures can arise, and then detail the devel-

opment of ONSTIM in the context of this generative model. In Chapter 4, we discuss the results of ONSTIM on synthetic datasets and on real financial datasets for a variety of parameter settings. Finally, in Chapter 5, we examine the strengths and weaknesses of ONSTIM, discuss avenues for further work, and provide concluding remarks.

## Background

In Chapter 2, we discuss background material relevant to the problem of inferring interaction structures from noisy observations. We first define and discuss graphical models, which are graphical representations of the conditional independence relationships between random variables in joint probability distributions. We then discuss algorithms for inference, some of which perform efficient inference by taking advantage of the aforementioned conditional independence relationships. We examine sampling algorithms, specifically Markov chain Monte Carlo (MCMC) algorithms, Metropolis-Hastings, and Gibbs sampling in some detail.

The second half of the background section is devoted to a detailed description of the work of Dzunic and Fisher [1], as their graphical model and inference algorithm form the core of the algorithm developed in this thesis. We first describe the graphical model used to represent the joint distribution of interest, and we then walk in substantial detail through the corresponding inference algorithm.

## ONSTIM

Chapter 3 contains the core of the work performed in this thesis. In this chapter, we first describe a generative model that proposes a mechanism by which new interaction structures can arise over some duration of time. We then detail the development of ONSTIM, focussing specifically on the setup that allows for online inference and on the initialization procedure during which new interaction structures are proposed.

## Results

In Chapter 4, we describe experimental results of ONSTIM on synthetic and real datasets. We describe the process by which synthetic datasets are generated, report results of the performance of ONSTIM in various parameter settings, and attempt to explain certain behaviors of ONSTIM from the results. We then apply ONSTIM to fi-

nancial datasets, consisting of one long term US equity dataset, one intraday US equity dataset, and one long term US sector ETF dataset.

### **Conclusion**

We conclude with a contextualization of the work performed in this thesis and a discussion of opportunities for improvement and augmentation of ONSTIM.



# Background

In this thesis, we are interested in the studying time-varying interaction structures among sets of time-varying signals. An interaction structure is an encoding of the of statistical dependence relationships among a set of signals. In order to introduce machinery to assist with reasoning over unconditional and conditional dependence relationships, we begin this chapter with a discussion of graphical models. We then consider various problems of inference that appear when studying graphical models and general joint probability distributions. Next, we introduce and describe the SSIM, a particular graphical model for describing time-varying interaction structures among a set of signals. Finally, we detail an algorithm for performing inference on the SSIM, which also forms the core of the new inference algorithm that we will present in the next chapter.

### ■ 2.1 Graphical Models

In the previous section, we discussed the importance of determining the structure of statistical dependence relationships among a set of random variables. Graphical models are concise representations of a family of joint distributions over a set of random variables that make evident the conditional dependence and independence relationships among them. A graphical model utilizes a graph to encode the dependences present in a set of random variables. Each node of the underlying graph represents a single random variable and each edge loosely represents a dependence between a pair of random variables. The exact interpretation of an edge in a graphical model, however, depends on whether the underlying graph is a undirected or directed. Such graphical models are referred to as undirected graphical models and directed graphical models respectively, and we describe both classes below.

### ■ 2.1.1 Undirected Graphical Models

First, we will discuss undirected graphical models, tools which provide a powerful model of conditional independence among sets of random variables. Let us consider an undirected graphical model with underlying graph  $g_t = \{V, E\}$ , where  $V$  is the set of vertices and  $E$  is the set of edges. Let  $A \subset V$ , and denote by  $p_A$  the joint distribution among the set of random variables represented by nodes in  $A$ . An undirected graphical model on  $g_t$  then describes the family of joint distributions that satisfy the following property:

$p_A$  and  $p_B$  are conditionally independent given  $p_C$  if and only if there exists no path from any node in  $A$  to any node in  $B_t$  that does not include a node in  $C$ .

While an undirected graphical model represents the conditional independence structure in a joint distribution, further parameterization of likelihoods and dependencies is necessary to actually specify the full distribution. The famous Hammersley-Clifford theorem establishes a link between the conditional independence property above and a parameterization of the joint distribution. Specifically, it states that a distribution that is positive everywhere satisfies the property above if and only if it can be written as

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \quad (2.1)$$

where  $\mathcal{C}$  is the set of all cliques in  $g_t$ ,  $x_C$  is the joint random variable on the nodes of  $C$ ,  $\psi_C$  is a positive function defined for all possible values of  $x_C$ , and  $Z$  is a normalization constant defined such that  $\sum_x p(x) = 1$  [4].

### ■ 2.1.2 Directed Graphical Models

In this thesis, we will concern ourselves primarily with directed graphical models, also known as Bayesian networks. Directed graphical models have directed graphs as the underlying structure, lending themselves better to an intuitive interpretation in terms of causality than in terms of conditional independence. Given a joint distribution  $p(x)$ , if the underlying directed graphical model is acyclic, then it is possible to write  $p(x)$  as the product of conditional distributions of each node given its parents. Specifically, we



can express  $p(x)$  as follows:

$$p(x) = \prod_{i=1}^N P(x_i | x_{pa(i)}). \quad (2.2)$$

This factorization lends itself to a very intuitive causality interpretation -  $x_i$  is causally affected by its parents.

Identifying conditional independence in directed graphical models is slightly more complicated. Complete conditions for independence are given by the Bayes' Ball algorithm, for whose details we refer the reader to [4]. A particular useful result of the Bayes' Ball algorithm is that a node is independent of all other nodes in the network conditioned on its children, parents, and children's parents, a subset of nodes called the original node's Markov blanket.

## ■ 2.2 Inference

In this section, we discuss some important problems in the field of inference. First, we discuss the belief propagation algorithm, an important algorithm for computing marginal distributions from a graphical model representation of a joint distribution. Next, we discuss the motivation behind obtaining samples from a joint distribution, and also algorithms for doing so that take advantage of graphical model structure. Finally, we will consider some families of conditional distributions that when coupled with specific prior distributions, allow for easy analytical computation of the posterior distribution.

### ■ 2.2.1 Belief Propagation

Given a joint distribution, an important problem is the computation of marginal distributions of a subset of the variables. In general, computing the marginal distribution of a subset of variables is computationally expensive. For example, suppose we have a joint distribution on  $N$   $k$ -valued random variables and we wish to compute the marginal distribution of  $M$  of them. Determining the marginal probability for each of the  $k^M$  possible values of the subset of  $M$  variables requires summing over all  $k^{N-M}$  possible values the remainder of the variables can take on, yielding a total cost of  $O(k^M \cdot k^{N-M}) = O(k^N)$ . However, the conditional independence information

present in a graphical model representation can be exploited to yield faster algorithms for marginalization.

Belief propagation is an algorithm for computing marginal distributions of specific random variables given a graphical model with parameters specifying the joint distribution. The core idea of marginalization with belief propagation is the notion that conditional independence between sets of variables reduces the total number of sums that must be computed for marginalization. A full implementation of belief propagation on a discrete distribution is given below in Algorithm 1. The runtime of Algorithm 1 is linear in the number of random variables in the graphical model [4].

```

Data:  $g_t, \phi, \psi$ 
Result:  $p(x_i) \forall i \in V$ 
 $D_t = \text{diameter}(g_t);$ 
for  $(i, j) \in E$  do
  |  $m_{i \rightarrow j}^0 = 1;$  // Initialize all messages to 1.
end
for  $t = 1 : 2D$  do
  | for  $(i, j) \in E$  do
  | |  $m_{i \rightarrow j}^t(x_j) = \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_{ij}) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^{t-1}(x_i);$  // Update messages.
  | end
end
for  $i \in V$  do
  |  $p(x_i) \propto \phi(x_i) \prod_{j \in N(i)} m_{j \rightarrow i}(x_i);$  // Compute marginal distribution.
end

```

**Algorithm 1.** Belief propagation algorithm for discrete variables.  $g_t$  is the graph underlying the graphical model,  $\phi$  is the set of node potentials, and  $\psi$  is the set of edge potentials.

Note that Algorithm 1 makes use of a summation to update messages. Often, however, we are interested in continuous distributions as well. In this case, we integrate over the support of  $x_i$  instead of computing a summation. Unfortunately, analytical computation of the integral

$$\int_{x_i} \phi_i(x_i) \psi_{ij}(x_{ij}) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^{t-1}(x_i)$$

is rarely tractable. One notable exception is the case of Gaussian graphical models, in which all node potentials, edge potentials, and therefore message take the form of Gaussian distributions. While belief propagation can be performed for any graph, it is only guaranteed to converge to the correct marginal distributions in the case of a graph with no cycles, i.e. a forest.

### ■ 2.2.2 Sampling

Often, we are interested in computing the expected value of some function  $f_t$  of a random variable  $X$  with joint distribution  $p_X$  that can take on values in  $\mathcal{X}$ . This expression is given by

$$\mathbb{E}_p[f_t(X)] = \sum_{x \in \mathcal{X}} p_X(x) f_t(x) \quad (2.3)$$

in the case of a discrete random variable, or

$$\mathbb{E}_p[f_t(X)] = \int_{x \in \mathcal{X}} p_X(x) f_t(x) dx \quad (2.4)$$

in the case of a continuous random variable. Even marginalization technically falls into this class of problem, as  $P(X_1 = x_1)$  can be equivalently written as  $\mathbb{E}_p[\mathbb{1}_{X_1=x_1}]$ , where  $X_1$  is one dimension of  $X$ . If  $X$  is discrete, brute force computation of  $\mathbb{E}_p[f_t(X)]$  requires summing over a number of terms exponential in the dimension of  $X$ . In the continuous case, this requires evaluating integrals that in general are not tractable. Since exact evaluation of  $\mathbb{E}_p[f_t(X)]$  is often either computationally or analytically intractable, we employ Monte Carlo methods to obtain an approximation.

Monte Carlo methods approach the problem of computing the expectation of a functions of a random variable by computing the function on samples taken from the joint distribution and then averaging. The theoretical grounding for Monte Carlo methods is based on the Strong Law of Large Numbers, which gives us

$$\frac{1}{N} \sum_{i=1}^N f_t(\hat{x}_i) \rightarrow \mathbb{E}_p[f_t(X)] \text{ as } N \rightarrow \infty, \text{ with probability 1,} \quad (2.5)$$

where the  $\hat{x}_i$  are independent samples taken from  $p_X$ . In order to employ this technique, we must be able to obtain a large number of independent samples from the joint distribution.

### Markov Chain Monte Carlo

Generating independent samples from a joint distribution in an efficient manner is not easy, and significant research has been devoted to this problem. Since direct sampling from  $p_X$  can be difficult hard, one approach is to use the  $p_X$  to construct a Markov chain (from which samples can be taken) whose stationary distribution is the target distribution from which we wish to obtain samples. After some initialization period, referred to as “burn-in”, samples taken from the Markov chain are close [in some sense] to samples taken from the target distribution.

We can describe a Markov chain with its transition matrix  $P$ , where  $P_{ij}$  is the probability of transition from state  $i$  to state  $j$ . The distribution  $p_X$  is a stationary distribution of  $P$  if

$$p_X(x) = \sum_{x'} p_X(x')P(x|x') \quad \forall x \in \mathcal{X}. \quad (2.6)$$

In the continuous case, we replace the transition matrix with a transition kernel and the sum above with an integral. A general Markov chain may have multiple stationary distributions, which is an undesirable quality in an MCMC algorithm, as this would provide no guarantee that samples were being taken from the correct stationary distribution.

We are thus interested in the construction of Markov chains that are guaranteed to have exactly one stationary distribution. This property is satisfied by a class of Markov chains called ergodic Markov chains, for whose precise definition we refer the reader to [10]. We are thus interested in algorithms to construct ergodic Markov chains with  $p_X$  as a stationary distribution.

### Metropolis-Hastings

Metropolis-Hastings is an algorithm for constructing the Markov chain  $P$  with the desired distribution  $p_X$  as its stationary distribution [5]. Since the target distribution  $p_X$  is difficult to sample from, Metropolis-Hastings operates by sampling from another conditional distribution,  $Q(\cdot|\cdot)$ , called the proposal distribution, which is easy to sample from but does not directly yield samples of the desired Markov chain  $P$ . The values of  $p_X$  at the previous sample of the Markov chain  $P$  and at the sample from  $Q$  are then used to determine whether to accept the new sample from  $Q$ .



In order to determine exactly how to construct the desired Markov chain  $P$  from the target distribution  $p_X$  and proposal distribution  $Q$ , we must introduce the concept of *detailed balance*. A Markov chain  $P$  satisfies detailed balance with respect to a target distribution  $p_X$  if

$$p_X(x)P(x'|x) = p_X(x')P(x|x'), \quad \forall x, x' \in \mathcal{X}. \quad (2.7)$$

If  $P$  satisfies detailed balance with respect to  $p_X$ , then  $p_X$  is a stationary distribution of  $P$  [cite source].

We can only obtain samples from  $Q$  while we desire samples from  $P$ , so we must modify samples from  $Q$  in a fashion that yields samples from  $P$ . This can be accomplished by accepting the sample from  $Q$  with a certain probability, and rejecting it otherwise. The probability of accepting the new sample  $x'$  from  $Q$  given the old sample  $x$  is known as the acceptance ratio, which we denote by  $a(x \rightarrow x')$ , and it is given by

$$a(x \rightarrow x') = \min\left\{1, \frac{p(x')Q(x|x')}{p(x)Q(x'|x)}\right\}. \quad (2.8)$$

The full conditional distribution of  $P(x'|x)$  is thus given by

$$P(x'|x) = \begin{cases} \min\{Q(x'|x), Q(x|x')\frac{p(x')}{p(x)}\} & \text{if } x' \neq x \\ Q(x|x) & \text{if } x' = x. \end{cases} \quad (2.9)$$

It is straightforward to show that the constructed  $P$  satisfies detailed balance with respect to the target distribution  $p_X$ , and thus sampling from  $P$  asymptotically yields samples from  $p_X$ .

Note that since values of  $p_X$  are only used in ratio form, Metropolis-Hastings allows for sampling from  $p_X$  even if the distribution is only known up to a constant factor in some unnormalized form  $\tilde{p}_X$ . The steps of Metropolis-Hasting are detailed below in Algorithm 2, assuming that the target distribution is provided in a general unnormalized form  $\tilde{p}_X(x)$ .

**Data:**  $\tilde{p}_X, Q$

**Result:**  $x_i \forall i \in \{0, \dots, T\}$

Initialize sample chain to  $x_0$ ;

**for**  $t = 1 : T$  **do**

Propose  $x'_t$  by sampling from  $Q(x'|x_{t-1})$ ;

Compute acceptance ratio  $a(x_{t-1} \rightarrow x'_t) = \min\{1, \frac{\tilde{p}(x'_t)Q(x_{t-1}|x'_t)}{\tilde{p}(x_{t-1})Q(x'_t|x_{t-1})}\}$ ;

Set  $x_t = \begin{cases} x'_t & \text{w.p. } a(x_{t-1} \rightarrow x'_t) \\ x_{t-1} & \text{w.p. } 1 - a(x_{t-1} \rightarrow x'_t) \end{cases}$  ;

**end**

**Algorithm 2.** Metropolis-Hastings.

### Gibbs Sampling

Although direct sampling from the full joint distribution of the vector valued random variable  $X$  can be intractable, it can happen that sampling from a conditional distribution of some subset of the dimensions given the remaining ones is possible. This is especially likely if the joint distribution of  $X$  can be expressed with a sparse graphical model containing many conditional independencies, as this allows for significant simplification of the conditional distributions.

Gibbs sampling is a special case of Metropolis-Hastings that takes advantage of the ease of sampling from conditional distributions to yield a very simple MCMC algorithm. At each iteration, a dimension of  $X$  is chosen at random, and the proposal distribution is taken to be the condition distribution of the chosen dimension given values of the previous sample in the remaining dimensions. It can be shown that Gibbs sampling always yields an acceptance ratio of 1, and since the resulting algorithm is guaranteed to satisfy detailed balance, Gibbs sampling constructs a simple Markov chain that yields samples from the target distribution.

As with all MCMC algorithms, Gibbs samplers require a burn-in period before samples are sufficiently close to being taken from the target distribution. Note that initialization of the Gibbs sampler, while unrelated to the eventual convergence of the sampler, can affect the burn-in period and therefore total convergence time. Apart from samples taken during the burn-in period, consecutive samples even from later on in the chain are clearly correlated, so several samples are often discarded between two that

are taken as true samples from the chain.

**Result:**  $x_i \forall i \in \{0, \dots, T\}$   
 $D_t = \text{dimension}(X)$ ;  
 Initialize sample chain to  $x_0$ ;  
**for**  $t = 1 : T$  **do**  
 | Sample  $i$  uniformly at random from  $\{1, \dots, D_t\}$ ;  
 | Sample  $x_t^i \sim P(x_t^i | x_{t-1}^{-i})$ ;  
 | Set  $x_t^{-i} = x_{t-1}^{-i}$ ;  
**end**

**Algorithm 3.** Gibbs sampling algorithm. Here, we use the notation  $x^{-i}$  to refer to all dimensions of  $x$  except for  $x^i$ .

### ■ 2.2.3 Conjugate Priors

Recall from the previous subsection that Gibbs sampling generates samples from a joint distribution by iteratively sampling from the posterior distribution of some subset of the variables given the remaining ones. In general, computation of the posterior distribution is not necessarily tractable, as it may require computing an intractable integral. In certain cases, however, a mathematical relationship between a conditional distribution and a special type of prior can yield a tractable closed form solution for the posterior.

Suppose that a random variable  $X$  representing data has a distribution parameterized by the random variable  $\Theta$ , yielding a conditional distribution of  $P(X|\Theta)$  which we will sometimes refer to as the likelihood model. Suppose furthermore that a prior distribution exists on  $\Theta$  that has a deterministic hyperparameter  $\gamma$ , which we write as  $P(\Theta; \gamma)$ . We wish to compute the posterior distribution of  $\Theta$  given  $X$ , perhaps to generate a conditional distribution from which to generate samples in a step of a Gibbs sampling procedure. We can approach the computation of this posterior using Bayes'

Rule:

$$P(\Theta|X) = \frac{P(X|\Theta)P(\Theta; \gamma)}{P(X; \gamma)} \quad (2.10)$$

$$= \frac{P(X|\Theta)P(\Theta; \gamma)}{\int_{\Theta} P(X|\Theta)P_0(\Theta; \gamma)d\Theta} \quad (2.11)$$

$$\propto P(X|\Theta)P(\Theta; \gamma). \quad (2.12)$$

For a general prior  $P(\Theta; \gamma)$  and likelihood model  $P(X|\Theta)$ , we cannot say anything about the form or parameterization of the posterior  $P(\Theta|X; \gamma)$ . However, certain classes of likelihood models can be coupled with priors called *conjugate priors* such that the posterior remains in the same family of distributions as the prior. We formalize this notion in the definition below.

**Definition 2.2.1.**  $P(\Theta; \gamma)$  is a conjugate prior to the likelihood model  $P(X|\Theta)$  if there exists a hyperparameter value  $\gamma'$  such that:

$$P(\Theta; \gamma') = P(\Theta|X; \gamma) \propto P(X|\Theta)P(\Theta; \gamma). \quad (2.13)$$

The existence of a conjugate prior for a likelihood model greatly simplified computation of the posterior. Instead of computing any complicated integral, the posterior can be determined by simply evaluating the posterior hyperparameter  $\gamma'$ , which can be expressed as a function of the original hyperparameter  $\gamma$  and the data  $X$ . Below we will detail certain pairs of likelihood model and conjugate prior distributions that are important for inference procedures in this thesis.

### Multinomial/Dirichlet

The multinomial distribution generalizes the binomial distribution to trials with more than two outcomes and the categorical distribution to multiple trials. It is parameterized by  $n$ , which is the number of trials, and by the vector of event probabilities  $\{\pi_i\}_{i=1}^K$ , where each trial has  $K$  possible outcomes and  $\sum \pi_i = 1$ . If  $Z \sim \text{Mult}(n, \pi)$ , then the pmf of  $Z$  is given by:

$$P(Z = z_1, \dots, z_K | \pi; n) = \frac{n!}{z_1! \dots z_K!} \pi_1^{z_1} \dots \pi_K^{z_K}. \quad (2.14)$$

The Dirichlet distribution is the conjugate prior for the multinomial distribution. The Dirichlet distribution is parameterized with hyperparameter  $\alpha = (\alpha_1, \dots, \alpha_K)$ ,



which correspond to pseudocount values. Its support is the  $K$ -dimensional simplex, consisting of the points  $(\pi_1, \dots, \pi_K)$  such that  $\sum \pi_i = 1$  and each  $\pi_i \in [0, 1]$ . Suppose  $\pi$  is distributed according to a Dirichlet distribution with hyperparameter  $\alpha$ . The density function of  $\pi$  is then given by:

$$P(\pi; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K \pi_i^{\alpha_i - 1}, \quad (2.15)$$

where the normalization constant  $B(\alpha)$  is given by:

$$B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}. \quad (2.16)$$

We will now show conjugacy of the Dirichlet prior to the multinomial distribution. Consider the posterior distribution  $P(\pi|Z; \alpha)$ . We can write this as:

$$P(\pi|Z; \alpha) \propto P(Z|\pi; n)P(\pi; \alpha) \quad (2.17)$$

$$\propto \pi_1^{z_1} \dots \pi_K^{z_K} \prod_{i=1}^K \pi_i^{\alpha_i - 1} \quad (2.18)$$

$$= \prod_{i=1}^K \pi_i^{\alpha_i + z_i - 1} \quad (2.19)$$

$$\propto \text{Dir}(\pi; \alpha_1 + z_1, \dots, \alpha_K + z_K) \quad (2.20)$$

$$= \text{Dir}(\pi; \alpha'), \quad (2.21)$$

where  $\alpha' = (\alpha_1 + z_1, \dots, \alpha_K + z_K)$ . Updating a Dirichlet prior given a multinomial likelihood model is thus simply tantamount to increasing the counts of  $\alpha$  by the number of observations from each category.

### Linear Gaussian/Matrix Normal-Inverse Wishart

The matrix normal distribution is a generalization of the multivariate normal to a support over some space of matrices. We will describe this distribution in terms of linear Gaussian model. Suppose that we wish to model the conditional distribution  $P(y|x)$ , where  $y \in \mathbb{R}^d$  and  $x \in \mathbb{R}^m$ . We can model the relationship between  $x$  and  $y$  as a linear Gaussian model, given by

$$y = Ax + \epsilon, \quad (2.22)$$

where  $A \in \mathbb{R}^{d \times m}$ , and  $\epsilon \in \mathbb{R}^d$  is drawn from a zero-mean multivariate normal distribution with covariance  $\Sigma$ . The conditional distribution of  $y$  given  $x$  can be parameterized by a parameter  $\Theta = (A, \Sigma)$ , allowing us to write the conditional likelihood model as:

$$P(y|x, \Theta) = P(y|x, A, \Sigma). \quad (2.23)$$

We are interested in characterizing a prior on  $(A, \Sigma)$  that is conjugate to the likelihood model above, a prior which is called the matrix-normal inverse Wishart distribution.

The matrix-normal inverse Wishart distribution is defined over  $A$  and  $\Sigma$  in a form that factors into an inverse-Wishart distribution over  $\Sigma$  and a matrix-normal distribution on  $A$  that is parameterized by the  $\Sigma$  sampled from the inverse-Wishart distribution. The distribution is given by:

$$\mathcal{MN}\mathcal{IW}(A, \Sigma; \Omega, \kappa, \Xi, \nu) = \mathcal{MN}(A; \Omega, \Sigma, \kappa) \mathcal{IW}(\Sigma; \Xi, \nu), \quad (2.24)$$

where the matrix-normal distribution on  $A$ , denoted by  $\mathcal{MN}(A; \Omega, \Sigma, \kappa)$ , is given by:

$$\mathcal{MN}(A; \Omega, \Sigma, \kappa) = \frac{|\kappa|^{d/2}}{|2\pi\Sigma|^{m/2}} \exp\left\{-\frac{1}{2}\text{Tr}[(A - \Omega)^T \Sigma^{-1}(A - \Omega)\kappa]\right\}, \quad (2.25)$$

and the inverse-Wishart distribution on  $\Sigma$ , denoted by  $\mathcal{IW}(\Sigma; \Xi, \nu)$ , is given by:

$$\mathcal{IW}(\Sigma; \Xi, \nu) = \frac{|\Xi|^{\nu/2} |\Sigma|^{-\frac{d+\nu+1}{2}} \exp\left\{-\frac{1}{2}\text{Tr}(\Xi\Sigma^{-1})\right\}}{2^{\frac{\nu d}{2}} \pi^{\frac{d(d-1)}{4}} \prod_{i=1}^d \Gamma\left(\frac{\nu+1-i}{2}\right)}, \quad (2.26)$$

with hyperparameters  $\Omega \in \mathbb{R}^{d \times m}$ ,  $\kappa \in \mathbb{R}^{m \times m}$ ,  $\Xi \in \mathbb{R}^{d \times d}$ , and  $\nu \in \mathbb{R}$ .

Suppose  $N$  observations of  $(x, y)$  pairs are taken from the linear Gaussian model parameterized by  $A$  and  $\Sigma$ . Let  $x_n$  and  $y_n$  denote the  $n^{\text{th}}$  observation of  $x$  and  $y$  respectively. The posterior update on the hyperparameters of the matrix is then given by:

$$\kappa' = \kappa + N, \quad (2.27)$$

$$\nu' = \nu + N, \quad (2.28)$$

$$\Omega' = \Sigma_{y,x} \Sigma_{x,x}^{-1}, \quad (2.29)$$

$$\Xi' = \Xi + \Sigma_{y|x}, \quad (2.30)$$

where

$$\Sigma_{x,x} = \Sigma_{n=1}^N x_n x_n^T + \kappa, \quad (2.31)$$

$$\Sigma_{y,x} = \Sigma_{n=1}^N y_n x_n^T + \Omega \kappa, \quad (2.32)$$

$$\Sigma_{y,y} = \Sigma_{n=1}^N y_n y_n^T + \Omega \kappa \Omega^T, \quad (2.33)$$

$$\Sigma_{y|x} = \Sigma_{y,y} - \Sigma_{y,x} \Sigma_{x,x}^{-1} \Sigma_{y,x}^T. \quad (2.34)$$

For more detail on the matrix-normal inverse-Wishart update, we refer the interested reader to [7].

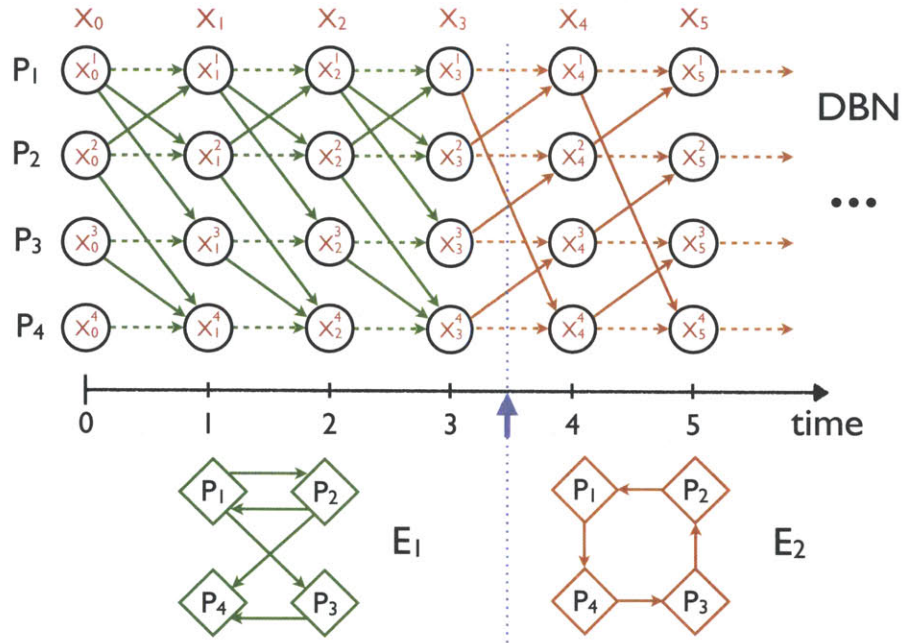
## ■ 2.3 Switching State-Space Interaction Model: Graphical Model

In this section, we will describe the switching state-space temporal interaction model (SSIM), graphical model developed by Dzunic and Fisher to model switching interaction structures between time series with noisy observations (see Figure 2.2). To motivate development of the model, recall the problem of learning the structure of a graphical model from data sampled from the joint distribution. In the context of SSIM, such structures are referred to as interaction structures, as they represent the interactions among the variables of interest.

SSIM allows for the inference of interaction structures that vary over time. Furthermore, the interaction structures are assumed to govern the behavior of unobserved latent variables, which yield observations through an observation model. In order to describe the evolution of interaction structures over time, we will first discuss dynamic Bayesian networks, generalizations of graphical models that allow for variables in the time domain. Next, we will detail linear Gaussian state space interaction models, which represent the evolution of the latent variable sequence in SSIM and its relationship to the observed data. Then, we will discuss certain conjugate priors, tools that are useful for analytical Bayesian reasoning with continuous variables. We will then conclude the section with a description of the SSIM graphical model.

### ■ 2.3.1 Dynamic Bayesian Networks

Dynamic Bayesian networks (DBN) adopt the notion of static interaction structures to models where the variables move through time. For each node in the graphical model representing the static interaction structure, a DBN consists of a sequence of nodes,



**Figure 2.1.** Dynamic Bayesian network (DBN) example. Figure obtained from [1] with permission of authors.

one for each time point in the model. If node  $X$  is a parent of node  $Y$  in the static interaction structure, then the DBN consists of an edge going from  $X_t$  to  $Y_{t+1}$  for all times  $t$ .

For example, suppose that we have a interaction structure consisting of four signals,  $P_1, P_2, P_3$ , and  $P_4$ , whose initial interaction structure  $E_1$  is shown in Figure 2.1. The corresponding DBN is shown above the interaction structure from times 0 to 3. Note that a DBN allows for a change in the static interaction structure over time, as shown in Figure 2.1 between time points 3 and 4. For purposes of tractability, we assume that each node has itself as a parent, an assumption that is typically reasonable in practice.

### ■ 2.3.2 Linear Gaussian State Space Interaction Model

A general state space model represents a system in terms of a set of input variables, a set of state variables that are not directly observable, and a set of observed variables that are derived from the state variables in a possibly stochastic fashion. State space models can represent systems that evolve in either continuous or discrete time, but in this thesis we will restrict ourselves to consideration of discrete time systems. Let us

represent the sets of input, state, and output variables as vectors which we name  $X$ ,  $u$ , and  $Y$  respectively. A general discrete-time state space model is determined by the functions  $f_t$  and  $g_t$ , respectively called the transition and observation models, as shown in the equations below:

$$X_t = f_t(X_{t-1}, u_t) \quad (2.35)$$

$$Y_t = g_t(X_t, u_t). \quad (2.36)$$

An important subclass of discrete-time state space models are the linear discrete-time state space models, in which  $f_t$  and  $g_t$  are both linear functions of their arguments. This is equivalent to expressing  $f_t$  and  $g_t$  in terms of matrix operations on their arguments, as shown below:

$$X_t = A_t X_{t-1} + B_t u_t \quad (2.37)$$

$$Y_t = C_t X_t + D_t u_t. \quad (2.38)$$

We will now restrict the input vector  $u_t$  to be only stochastic, i.e. zero-mean noise. Furthermore, we will restrict the matrices  $B_t$  and  $D_t$  such that any row index that corresponds to a nonzero row in  $B_t$  must correspond to a zero row in  $D_t$ , and any row index that corresponds to a nonzero row in  $D_t$  must correspond to a zero row in  $B_t$ . By doing so, we have effectively decoupled  $u_t$  into two subvectors, one that only influences the transition model, and one that only influences the observation model. Denote the transition subvector by  $u_{t,1}$  and the observation subvector by  $u_{t,2}$ , and define  $\epsilon_t = B_t u_{t,1}$  and  $\epsilon_{t,obs} = D_t u_{t,2}$ . Our restricted linear state space model is now described by:

$$X_t = A_t X_{t-1} + \epsilon_t \quad (2.39)$$

$$Y_t = C_t X_t + \epsilon_{obs,t}. \quad (2.40)$$

Suppose now that  $\epsilon_t$  and  $\epsilon_{obs,t}$  both have multivariate Gaussian distributions with mean zero and covariance matrices  $\Sigma_t$  and  $\Sigma_{obs,t}$  respectively. Since the family of multivariate Gaussian distributions is closed under linear combinations, taking  $X_0$  to be distributed according to a multivariate Gaussian as well yields multivariate Gaussian distributed  $X_t$  and  $Y_t$ , for all  $t \geq 0$ . So far, we have described what is called a linear Gaussian state space model.

Finally, we wish to impose the notion of interaction structure described above onto the linear Gaussian state space model. Let the parent set of the  $i^{\text{th}}$  entry of  $X$  (called



$X^i$ ) under the interaction structure active at time  $t$  be denoted by  $\tilde{pa}(i, t)$ . In this thesis, we will restrict the  $i^{\text{th}}$  entry of  $Y$ , written as  $Y^i$ , to depend only on  $X^i$  and observation noise. Furthermore, we will restrict the number of possible observation models to 1, thereby allowing no time variation of the observation model. We can thus write our final linear Gaussian state space interaction model as a collection of models for each element of  $X$ , shown below for  $X^i$ :

$$X_t^i = A_t^i X_{t-1}^{\tilde{pa}(i,t)} + \epsilon_t^i, \quad \epsilon_t^i \sim \mathcal{N}(0, \Sigma_t^i) \quad (2.41)$$

$$Y_t^i = C^i X_t^i + \epsilon_{obs}^i, \quad \epsilon_{obs}^i \sim \mathcal{N}(0, \Sigma_{obs}^i). \quad (2.42)$$

We will often write the joint parameters of the transition model as  $\theta_t^i = (A_t^i, \Sigma_t^i)$  and of the observation model as  $\xi^i = (C^i, \epsilon_{obs}^i)$ . Note that we can also interpret the above equations as specifying the conditional distributions of  $X_t^i$  and  $Y_t^i$  as follows:

$$P(X_t^i | X_{t-1}^{\tilde{pa}(i,t)}, \theta_t^i) = \mathcal{N}(X_t^i; A_t^i X_{t-1}^{\tilde{pa}(i,t)}, \Sigma_t^i) \quad (2.43)$$

$$P(Y_t^i | X_t^i, \xi^i) = \mathcal{N}(Y_t^i; C^i X_t^i, \Sigma_{obs}^i). \quad (2.44)$$

### ■ 2.3.3 Graphical Model

We finally turn to describing the graphical model of SSIM, shown in Figure 2.2. Suppose we wish to perform inference over the interaction structures among  $N$  objects as the system evolves from time  $t = 0$  to  $T$ . In general, each of the  $N$  objects can have any subset of the  $N$  objects as a parent set. Since there exist  $2^N$  possible parent sets for each of the  $N$  objects, the total number of possible interaction structures over  $N$  components is given by  $(2^N)^N = 2^{N^2}$ , which is superexponential in the number of objects. In order to restrict ourselves to a tractable number of possible interaction structures to reason over, we limit the maximum number of parents any node can have to  $M$ ,

Let  $X_t^i$  denote the state of the  $i^{\text{th}}$  object at time  $t$ , let  $E_t$  denote the interaction structure active at time  $t$ , and let  $\theta_t$  denote the parameters of the transition model active at time  $t$ . Recall that an interaction structure in a dynamic Bayesian network consists of a set of parents from the previous time point for each node. Given an interaction structure and transition model parameters, we can write the distribution of  $X_t^i$  as  $P(X_t^i | X_{t-1}^{pa(i,t)}, \theta_t^i)$ , where  $pa(i, t)$  is the parent set of  $X_t^i$  given by  $E_t$ , and where  $\theta_t^i$  is the parameter of the transition model for object  $i$ . We assume independence of transition models across all  $N$  objects, so we can write the full distribution of  $X_t$

conditioned on  $X_{t-1}$ , interaction structure, and parameters as:

$$P(X_t|X_{t-1}, E_t, \theta_t) = \prod_{i=1}^N P(X_t^i|X_{t-1}^{pa(i,t)}, \theta_t^i). \quad (2.45)$$

Given the graphical model structure of SSIM, we can write the full distribution of  $X$  as:

$$P(X_t|X_{t-1}, E_t, \theta_t) = \prod_{i=1}^N P(X_t^i|X_{t-1}^{pa(i,t)}, \theta_t^i). \quad (2.46)$$

Next, suppose that the active interaction structure and transition model parameters at any point in time comes from one of  $K$  available structure/parameter pairs. Let  $Z_t \in \{1, \dots, K\}$  denote the index of the structure/parameter model at time  $t$ . When indexed by the model number instead of the time, the interaction structure and model parameters are written with a tilde. We can express this equivalence as  $E_t = \tilde{E}_{Z_t}$  and  $\theta_t = \tilde{\theta}_{Z_t}$ . Thus, we can rewrite the distribution of  $X_t$  as:

$$P(X_t|X_{t-1}, E_t, \theta_t) = P(X_t|X_{t-1}, Z_t, \tilde{E}, \tilde{\theta}) \quad (2.47)$$

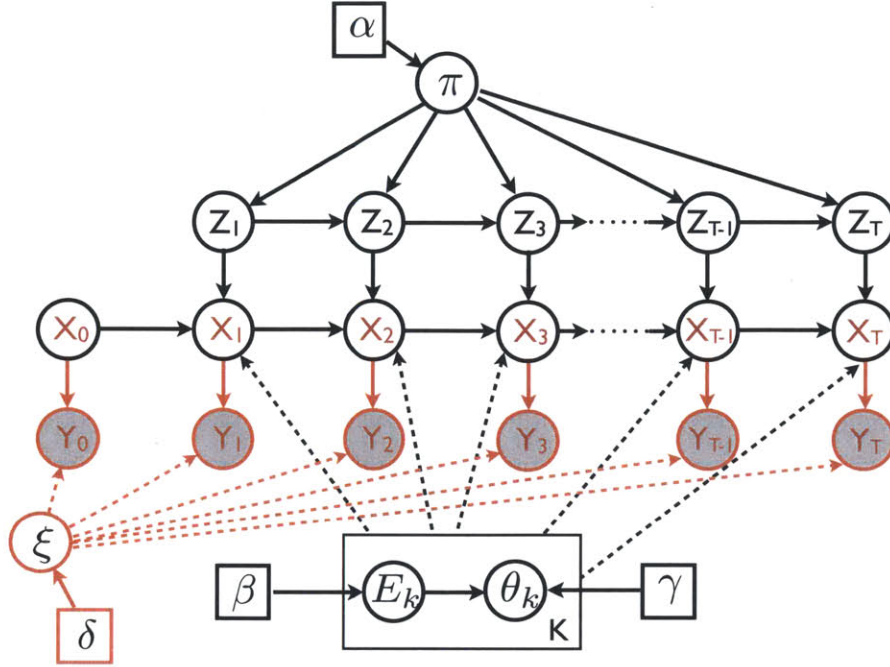
$$= P(X_t|X_{t-1}, \tilde{E}_{Z_t}, \tilde{\theta}_{Z_t}) \quad (2.48)$$

$$= \prod_{i=1}^N P(X_t^i|X_{t-1}^{pa(i,Z_t)}, \tilde{\theta}_{Z_t}^i). \quad (2.49)$$

We refer to  $Z$  as the switching sequence, and to the value of  $Z_t$  as the switching state at time  $t$ . We impose a first order Markov chain on the switching states, with initial and transition probabilities given by the multinomial distributions of  $\pi$ , which we detail below.

Finally, we assume a linear Gaussian state space model on the latent state trajectory  $X$  and the observed sequence  $Y$ . The transitions of  $X$  according to the model has already been described above. The observation model, parameterized by  $\xi$ , describes the dependence of  $Y$  on  $X$ . We assume that each entry  $i$  in  $Y_t$ , which we denote by  $Y_t^i$ , is dependent only on  $X_t^i$  and the noise model for object  $i$ , which we denote by  $\xi^i$ . Thus, we can write the distribution of  $Y_t^i$  as:

$$P(Y_t|X_t, \xi) = \prod_{i=1}^N P(Y_t^i|X_t^i, \xi^i). \quad (2.50)$$



**Figure 2.2.** State-space switching interaction model (SSIM). Figure obtained from [1] with permission of authors.

The relationships between the variable sequences  $X$ ,  $Y$ , and  $Z$  are governed by the parameters  $\tilde{E}$ ,  $\tilde{\theta}$ ,  $\pi$ , and  $\xi$ . These dependencies are depicted in the SSIM graphical model (Figure 2.2). Next, we detail the parameter variables  $\tilde{E}$ ,  $\tilde{\theta}$ ,  $\pi$ , and  $\xi$  some more, paying particular attention to their prior distributions.

### Interaction Structures and Transition Model Parameters: $\tilde{E}, \tilde{\theta}$

The variables  $\{\tilde{E}, \tilde{\theta}\}_{1:K}$  are a set of  $K$  interaction structures and transition model parameter sets, exactly one of which can be active at any time. Here, we will discuss the distribution of a single pair of an interaction structure and its transition model parameters, which we will denote here as  $\tilde{E}, \tilde{\theta}$ , and which we will commonly refer to as a structure-parameter pair. While  $\tilde{E}$  and  $\tilde{\theta}$  are written as separate random variables, the two variables are intimately coupled, as the very support of  $\tilde{\theta}$  is dependent on the value of  $\tilde{E}$ . Note that  $\tilde{E}$  and  $\tilde{\theta}$  are the interaction structures and transition model parameters respectively from our discussion of linear Gaussian state space interaction models.



Recall that an interaction structure  $\tilde{E}$  is determined by a vector of a specific parent set for each object in  $X$ . We adopt a prior on  $\tilde{E}$ , called the structural prior, that is parameterized by the hyperparameter vector  $\beta$ , which contains a scalar value for each object-parent set pair. The prior probability for any structure assumes a modular prior on structure, and is therefore proportional to the product of these scalar values for each object-parent set pair in the structure, as shown here:

$$P(\tilde{E}; \beta) = \frac{1}{Z(\beta)} \prod_{i=1}^N \beta_{i, \tilde{p}a(i)} \propto \prod_{i=1}^N \beta_{i, \tilde{p}a(i)}, \quad (2.51)$$

where  $Z(\beta)$  is a normalization constant chosen so that  $\sum_{\tilde{E}} P(\tilde{E}; \beta) = 1$ . Recall that since the total number of permissible interaction structures is polynomial in  $N$ , evaluation of the  $Z(\beta)$  is computationally tractable.

The parameter set  $\tilde{\theta}$  is a collection of  $N$  random variables, one for each object-parent set pair present in the interaction structure  $\tilde{E}$ . We will now detail the prior on the distribution of  $\tilde{\theta}$  given  $\tilde{E}$  and the hyperparameter  $\gamma$ . Like  $\beta$ ,  $\gamma$  is a vector that contains an entry for each possible object-parent set pair. Unlike  $\beta$  which contains a scalar for each such pair, however,  $\gamma$  contains a matrix-normal inverse-Wishart prior for each object-parent set pair. First, we assume parameter independence across objects, giving us the following decomposition of the full prior on  $\tilde{\theta}$ :

$$P(\tilde{\theta} | \tilde{E}; \gamma) = \prod_{i=1}^N P(\tilde{\theta}^i | \tilde{E}; \gamma_i). \quad (2.52)$$

Second, we assume that the prior probability on parameters for a given object  $i$  is a function only of the entries of  $\gamma$  that correspond to the parent set of  $i$  in  $\tilde{E}$ . This assumption, which we call parameter modularity, is given by:

$$P(\tilde{\theta}^i | \tilde{E}; \gamma_i) = P(\tilde{\theta}^i; \gamma_{i, \tilde{p}a(i)}). \quad (2.53)$$

Recall from our discussion of linear Gaussian state space interaction models that  $\theta^i$  is simply a tuple of a transition matrix and noise covariance matrix, which we write as  $\theta^i = (A^i, \Sigma^i)$ . We can now write the prior in matrix-normal inverse-Wishart form:

$$P(\tilde{\theta}^i; \gamma_{i, \tilde{p}a(i)}) = P(A^i, \Sigma^i; \gamma_{i, \tilde{p}a(i)}) \quad (2.54)$$

$$= \mathcal{MNIW}(A^i, \Sigma^i; M^{i, \tilde{p}a(i)}, \Omega^{i, \tilde{p}a(i)}, \kappa^{i, \tilde{p}a(i)}, \Psi^{i, \tilde{p}a(i)}) \quad (2.55)$$

$$= \mathcal{MN}(A^i, \Sigma^i; M^{i, \tilde{p}a(i)}, \Omega^{i, \tilde{p}a(i)}, \Sigma^i) \mathcal{IW}(\Sigma^i; \kappa^{i, \tilde{p}a(i)}, \Psi^{i, \tilde{p}a(i)}). \quad (2.56)$$

Thus, the full prior on coupled interaction structure and parameter set  $\tilde{E}, \tilde{\theta}$  is given by:

$$P(\tilde{E}, \tilde{\theta}; \beta, \gamma) = P(\tilde{E}; \beta)P(\tilde{\theta}|\tilde{E}; \gamma) \quad (2.57)$$

$$\propto \prod_{i=1}^N \beta_{i, \tilde{p}a(i)} \mathcal{MNTW}(A^i, \Sigma^i; M^{i, \tilde{p}a(i)}, \Omega^{i, \tilde{p}a(i)}, \kappa^{i, \tilde{p}a(i)}, \Psi^{i, \tilde{p}a(i)}). \quad (2.58)$$

### Discrete Markov Switching State Model: $\pi$

The discrete Markov switching state model  $\pi$  governs the transition dynamics of  $Z$  in a fashion similar to how  $\tilde{E}$  and  $\tilde{\theta}$  govern the transition dynamics of  $X$ .  $\pi$  consists of an initial multinomial distribution  $\pi_0$ , and a collection of  $K$  transition multinomial distributions  $\{\pi_1, \dots, \pi_K\}$ . The initial multinomial  $\pi_0$  defines the distribution of  $Z_1$  given  $\pi$ , such that:

$$P(Z_1 = z_1 | \pi) = \pi_{0, z_1}, \quad (2.59)$$

where  $\pi_{0, z_1}$  denotes the  $z_1^{\text{st}}$  entry of  $\pi_0$ . Similarly, a transition multinomial  $\pi_i$  defines the conditional distribution of  $Z_t$  given  $Z_{t-1}$  and  $\pi$  such that:

$$P(Z_t = z_t | Z_{t-1}, \pi) = \pi_{Z_{t-1}, z_t}. \quad (2.60)$$

Like  $\tilde{E}$  and  $\tilde{\theta}$ ,  $\pi$  also has a conjugate prior, which in the case of a multinomial distribution is a Dirichlet distribution. Since  $\pi$  is not just a single multinomial distribution but actually a collection of  $K + 1$  multinomial distributions, the prior on  $\pi$  is a collection of  $K + 1$  Dirichlet distributions. The prior on  $\pi_i$  is given by:

$$P(\pi_i; \alpha_i) = \text{Dir}(\pi_{i,1}, \dots, \pi_{i,K-1}; \alpha_{i,1}, \dots, \alpha_{i,K}). \quad (2.61)$$

### Observation Model: $\xi$

The observation model variable  $\xi$  is similar in nature to the parameter set variable  $\tilde{\theta}$ .  $\xi$  consists of a collection of variables  $\xi^i$  for each entry in the observation vector  $Y$ . We can write  $\xi^i$  as the tuple  $(C^i, \Sigma_{obs}^i)$ , where these variables determine the observation model dynamics in the linear Gaussian state space interaction model, as shown in Equation []. In this thesis, we will typically fix  $C^i$ , to be a constant, most often 1, leaving us  $\xi^i = \Sigma_{obs}^i$ . Thus, the prior of  $\xi^i$  is given by an inverse-Wishart distribution with prior

$\delta^i$ . Assuming independence of each entry of  $\xi$ , we can write the full prior over  $\xi_i$  as:

$$P(\xi; \delta) = \prod_{i=1}^N P(\xi^i; \delta^i) \quad (2.62)$$

$$= \prod_{i=1}^N \mathcal{IW}(\xi^i; \kappa_{obs}^i, \Psi_{obs}^i). \quad (2.63)$$

## ■ 2.4 Switching State-Space Temporal Interaction Model: Inference

Dzunic and Fisher performing inference on the switching state-space interaction model by obtaining samples from the joint distribution of  $(X, Z, \{\tilde{E}, \tilde{\theta}\}, \pi, \xi)$  using a Gibbs sampling approach. The high level Gibbs sampling algorithm is given below. Details of each of the steps of the Gibbs sampler along with the initialization method are covered in the subsection below.

**Data:**  $Y_{0:T}$   
**Result:**  $\text{Sample}_i \forall i \in \{0, \dots, T\}$   
Initialize sample chain;  
**for**  $t = 1$  : *number of samples* **do**  
    Sample  $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ ;  
    Sample  $Z \sim P(Z|X, \tilde{\theta}, \tilde{E}, \pi)$ ;  
    Sample  $\pi \sim P(\pi|Z; \alpha)$ ;  
    Sample  $\tilde{E}, \tilde{\theta} \sim P(\tilde{E}, \tilde{\theta}|Z, X; \beta, \gamma)$ ;  
    Sample  $\xi \sim P(\xi|X, Y; \delta)$ ;  
**end**

**Algorithm 4.** SSIM Gibbs sampler.

### ■ 2.4.1 Sample $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$

Samples of  $X$  are obtained jointly using a backwards message-passing and forwards sampling algorithm. Note that we can factor the conditional distribution of  $X$  as:

$$P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi) = P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) \prod_{t=1}^T P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi) \quad (2.64)$$

$$= P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) \prod_{t=1}^T P(X_t|X_{t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi), \quad (2.65)$$

where in the second equation, we made use of the Markov property of the SSIM directed graphical model.

Although each factor in the expression above only expresses a dependency between adjacent time points of  $X$ , each time point of  $X$  has a dependency on all of  $Y$ . In order to incorporate this dependency, we employ a message-passing algorithm. The first phase of the algorithm involves passing messages back from time  $T$  up to time 0, which provide information from future observations to a node at time  $t$ . Afterwards, the messages, observations, and previous values of  $X$  are used to sample  $X$  going forwards. The algorithm for sampling  $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$  is given below, and the specialization of message-passing and sampling for the case of a linear Gaussian state-space model is detailed afterwards.

```

Data:  $Z, Y, \tilde{E}, \tilde{\theta}, \xi$ 
Result:  $X$ 
 $m^T(x_T) = 1;$ 
// Computation of Backwards Messages
for  $t = T - 1 : 0$  do
|  $m^t(x_t) = \int_{X_{t+1}} P(X_{t+1}|x_t, E_{Z_{t+1}}, \Theta_{Z_{t+1}}) P(Y_{t+1}|X_{t+1}) m^{t+1}(X_{t+1}) dX_{t+1};$ 
end
// Forward Sampling
Compute  $P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi) \propto P(X_0) P(Y_0|X_0, \xi) m^0(X_0);$ 
Sample  $X_0 \sim P(X_0|Z, Y, \tilde{E}, \tilde{\theta}, \xi);$ 
for  $t = 1 : T$  do
| Compute  $P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi) \propto P(X_t|X_{t-1}, E_{Z_t}, \Theta_{Z_t}) P(Y_t|X_t, \xi) m^t(X_t);$ 
| Sample  $X_t \sim P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}, \xi);$ 
end

```

**Algorithm 5.** Sampling of  $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ .

### Computation of Gaussian Messages

Since we are operating in a linear Gaussian state space model, exact computation of the integral detailed in the backwards message-passing step of Algorithm 5 is tractable. The linear Gaussian model implies that all transition probabilities  $P(X_t|X_{t-1}, E_{Z_t}, \Theta_{Z_t})$  and all observation probabilities  $P(Y_t|X_t, \xi)$  take the form of Gaussian distributions. Since

the message  $m^T(x_T) = 1$  can be considered to be a Gaussian with infinite covariance, computation of  $m^{T-1}(X_{T-1})$  requires integrating over the product of three Gaussian factors, which yields a Gaussian distribution. Inductively, all messages passed backwards can be written as a Gaussian distribution parameterized by a mean vector and covariance matrix.

Here, we detail the computation of the Gaussian messages, which we express in a mean-covariance parameterization. Let  $m_T(x_T) = 1$ , which is equivalent to setting  $\mu_T^m = 0$  and  $(\Sigma_T^m)^{-1} = 0$ . Let  $(A_{Z_t}, \Sigma_{Z_t}) = \Theta_{Z_t}$  denote the transition model and noise covariance operational at time  $t$ , and let  $\Sigma_{obs} = \xi$  represent the observation model noise covariance. Then, for  $t \in \{T-1, \dots, 0\}$ , we recursively define the message mean  $\mu_t^m$  and message covariance  $\Sigma_t^m$  as shown below, in which we adapt notation from [1]:

$$B_t = \Sigma_{Z_{t+1}}^{-1} A_{Z_{t+1}} \quad (2.66)$$

$$\mu_t^* = \Sigma_{obs}^{-1} Y_{t+1} + \Sigma_{t+1}^{m-1} \mu_{t+1}^m \quad (2.67)$$

$$\Sigma_t^* = (\Sigma_{Z_{t+1}}^{-1} + \Sigma_{obs}^{-1} + \Sigma_{t+1}^{m-1})^{-1} \quad (2.68)$$

$$\Sigma_t^m = (B_t^T (\Sigma_{Z_{t+1}} - \Sigma_t^*) B_t)^{-1} \quad (2.69)$$

$$m^t(x_t) = \mathcal{N}(x_t; \mu_t^m, \Sigma_t^m). \quad (2.70)$$

The process of forwards sampling requires sampling from a product of Gaussians, which yields a Gaussian distribution. First, we detail the sampling of  $X_0$ , which involves sampling from the product of two Gaussians:

$$\Sigma_0'^{-1} = \Sigma_{obs}^{-1} + \Sigma_0^{m-1} \quad (2.71)$$

$$\mu_0' = \Sigma_0' (\Sigma_{obs}^{-1} Y_0 + \Sigma_0^{m-1} \mu_0^m) \quad (2.72)$$

$$X_0 \sim \mathcal{N}(x_0; \mu_0', \Sigma_0'). \quad (2.73)$$

Obtaining subsequent conditional samples of  $X_t$  requires sampling from the product of three Gaussian factors. Here, we detail the recursive sampling of  $X_t$  given  $X_{t-1}$ ,  $Z_t$ ,  $Y_t$ , and  $m^t(x_t)$ :

$$\Sigma_t'^{-1} = \Sigma_{Z_t}^{-1} + \Sigma_{obs}^{-1} + \Sigma_t^{m-1} \quad (2.74)$$

$$\mu_t' = \Sigma_t' (\Sigma_{Z_t}^{-1} A_{Z_t} X_{t-1} + \Sigma_{obs}^{-1} Y_t + \Sigma_t^{m-1} \mu_t^m) \quad (2.75)$$

$$X_t \sim \mathcal{N}(X_t; \mu_t', \Sigma_t'). \quad (2.76)$$

Altogether, these steps comprise a method for obtaining a joint sample of  $X_{0:T}$  conditional on  $Z, Y, \tilde{E}$ , and  $\tilde{\theta}$  in a linear Gaussian state space model.

### ■ 2.4.2 Sample $Z \sim P(Z|X, \tilde{E}, \tilde{\theta}, \pi)$

Next, we are interested in sampling the discrete switching sequence  $Z$  given  $X, \tilde{E}, \tilde{\theta}$ , and  $\pi$ . The sampling procedure, which consists of a backwards message-passing step followed by forwards sampling, is remarkably similar to the procedure for sampling  $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta}, \xi)$ , due to the analogous state-space model structure. Here, we will factor the conditional distribution of  $Z$  as:

$$P(Z|X, \tilde{E}, \tilde{\theta}, \pi) = P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi) \prod_{t=2}^T P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \xi) \quad (2.77)$$

$$= P(Z_1|Y, \tilde{E}, \tilde{\theta}, \pi) \prod_{t=2}^T P(Z_t|Z_{t-1}, X, \tilde{E}, \tilde{\theta}, \pi). \quad (2.78)$$

Unlike in the case of sampling  $X$ , however, we deal entirely with discrete distributions and messages here, as  $Z$  is a discrete-valued random variable. The algorithm for sampling  $Z$  is given below.

```

Data:  $X, \tilde{E}, \tilde{\theta}, \pi$ 
Result:  $Z$ 
 $m^T(z_T) = 1;$ 
// Computation of Backwards Messages
for  $t = T - 1 : 1$  do
|  $m^t(z_t) = \sum_{Z_{t+1}} P(Z_{t+1}|z_t, \pi) P(X_{t+1}|X_t, E_{Z_{t+1}}, \Theta_{Z_{t+1}}) m^{t+1}(Z_{t+1});$ 
end
// Forward Sampling
Compute  $P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi) \propto P(Z_1|\pi) P(X_1|X_0, E_{Z_1}, \Theta_{Z_1}) m^1(Z_1);$ 
Sample  $Z_1 \sim P(Z_1|X, \tilde{E}, \tilde{\theta}, \pi);$ 
for  $t = 1 : T$  do
| Compute  $P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi) \propto P(Z_t|Z_{t-1}, \pi) P(X_t|X_{t-1}, E_{Z_t}, \Theta_{Z_t}) m^t(Z_t);$ 
| Sample  $Z_t \sim P(Z_t|Z_{1:t-1}, X, \tilde{E}, \tilde{\theta}, \pi);$ 
end

```

**Algorithm 6.** Sampling  $Z \sim P(Z|X, \tilde{E}, \tilde{\theta}, \pi)$ .

### ■ 2.4.3 Sample $\tilde{E}, \tilde{\theta} \sim P(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma)$

Sampling the structure/parameter models  $\tilde{E}, \tilde{\theta}$  is arguably the core step in the Gibbs sampler for SSIM. Since each of the  $K$  models has an independent prior and separate likelihood models, so we can decompose the posterior distribution of  $\tilde{E}, \tilde{\theta}$  as follows:

$$P(\tilde{E}, \tilde{\theta} | Z, X; \beta, \gamma) = \prod_{k=1}^K P(\tilde{E}_k, \tilde{\theta}_k | \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta, \gamma). \quad (2.79)$$

Next we decompose the posterior distribution of a single structure/model based on the parent sets in the interaction structure:

$$P(\tilde{E}_k, \tilde{\theta}_k | \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta, \gamma) \quad (2.80)$$

$$= \prod_{i=1}^N P(\tilde{p}a(i, k), \tilde{\theta}_k^i | \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta, \gamma) \quad (2.81)$$

$$= \prod_{i=1}^N P(\tilde{p}a(i, k) | \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta) P(\tilde{\theta}_k^i | \tilde{p}a(i, k), \{X_t, X_{t-1}\}_{t:Z_t=k}; \gamma). \quad (2.82)$$

We will now consider the computation of the posterior distribution of parameter sets. Recall that a different parameter set exists for each object-parent set pair. Thus, we can write:

$$P(\tilde{\theta}_k^i | \tilde{p}a(i, k), \{X_t, X_{t-1}\}_{t:Z_t=k}; \gamma) = P(\tilde{\theta}_k^{i, \tilde{p}a(i, k)} | \{X_t^i, X_{t-1}^{\tilde{p}a(i, k)}\}_{t:Z_t=k}; \gamma), \quad (2.83)$$

where this update can be performed analytically by updating the hyperparameter  $\gamma$  due to the conjugacy of the multivariate normal likelihood model of  $P(X_t^i | X_{t-1}^{\tilde{p}a(i, k)}, \tilde{\theta}_k^{i, \tilde{p}a(i, k)})$  with the matrix-normal inverse-Wishart prior distribution of  $P(\tilde{\theta}_k^{i, \tilde{p}a(i, k)}; \gamma)$ . Next, we will consider the posterior distribution on parent sets. We apply algebraic manipulations to obtain an expression for the posterior distribution of parent sets in terms of the prior:

$$P(\tilde{p}a(i, k) | \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta) \quad (2.84)$$

$$\propto P(\tilde{p}a(i, k), \{X_t, X_{t-1}\}_{t:Z_t=k}; \beta) \quad (2.85)$$

$$= P(\tilde{p}a(i, k); \beta) P(\{X_t, X_{t-1}\}_{t:Z_t=k} | \tilde{p}a(i, k); \beta) \quad (2.86)$$

$$= P(\tilde{p}a(i, k); \beta) \prod_{t:Z_t=k} P(X_t^i | X_{t-1}^{\tilde{p}a(i, k)}; \gamma). \quad (2.87)$$

Finally, we must compute  $\prod_{t:Z_t=k} P(X_t^i | X_{t-1}^{\tilde{p}a(i, k)}; \gamma)$ , which is likelihood of  $X_t^i$  with parent set  $\tilde{p}a(i, k)$  at all times  $t$  indexed with model  $k$ , with all parameters marginalized

out. We can write this as:

$$\prod_{t:Z_t=k} P(X_t^i | X_{t-1}^{\bar{p}a(i,k)}; \gamma) \quad (2.88)$$

$$= \int_{\tilde{\theta}_k^{i,\bar{p}a(i,k)}} P(\tilde{\theta}_k^{i,\bar{p}a(i,k)}; \gamma) \left[ \prod_{t:Z_t=k} P(X_t^i | X_{t-1}^{\bar{p}a(i,k)}, \tilde{\theta}_k^{i,\bar{p}a(i,k)}; \gamma) \right] d\tilde{\theta}_k^{i,\bar{p}a(i,k)}. \quad (2.89)$$

Computation of the above integral is analytically feasible, once more due to the conjugacy of the likelihood model of  $P(X_t^i | X_{t-1}^{\bar{p}a(i,k)}, \tilde{\theta}_k^{i,\bar{p}a(i,k)}; \gamma)$  with the prior distribution of  $P(\tilde{\theta}_k^{i,\bar{p}a(i,k)}; \gamma)$ . After updating both  $\beta$  and  $\gamma$ , new structure/parameter models can be sampled directly from the posterior.

#### ■ 2.4.4 Sample $\pi \sim P(\pi | Z; \alpha)$

Sampling the discrete state transition model  $\pi$  given  $Z$  simply requires updating the hyperparameter  $\alpha$  due to conjugacy of the Dirichlet prior with a multinomial likelihood model. Let  $N_{i,j}$  denote the number of times that  $Z_{t-1} = i$  and  $Z_t = j$ , and let  $N_{0,j}$  denote the number of times that  $Z_1 = j$ . Then, for each multinomial  $\pi_i$ , we update the hyperparameter of the Dirichlet prior  $\alpha_i = (\alpha_{i,1}, \dots, \alpha_{i,K})$  to

$$\alpha'_i = (\alpha_{i,1} + N_{i,1}, \dots, \alpha_{i,K} + N_{i,K}). \quad (2.90)$$

We can then sample  $\pi$  directly from  $P(\pi; \alpha')$ .

#### ■ 2.4.5 Sample $\xi \sim P(\xi | X, Y; \delta)$

The sampling  $\xi \sim P(\xi | X, Y; \delta)$  is similar to sampling  $\tilde{\theta}$ . We have independence of observation models across entries of  $Y$ , so can write:

$$P(\xi | X, Y; \delta) = \prod_{i=1}^N P(\xi^i | X^i, Y^i; \delta^i). \quad (2.91)$$

The computation of  $P(\xi^i | X^i, Y^i; \delta^i)$  is analytically tractable, due to the conjugacy of the multivariate Gaussian likelihood model to the normal inverse-Wishart prior on  $\xi$ . We must simply update  $\delta$  and sample from  $\xi$  from the posterior parameterized by  $\delta'$ .

#### ■ 2.4.6 Initialization

In order to initialize the model, the transition model for every object at every time is set to have the identity matrix as the transition matrix, and a covariance matrix drawn from



a prior specified in the algorithm. Backwards message-passing and forwards sampling is then performed on  $X$  to obtain an initial sample of  $X$  given the fixed transition models and the observed sequence  $Y$ .

A discrete transition model  $\pi$  is then sampled from the prior parameterized by  $\alpha$ . The switching sequence  $Z$  is then initialized according to the sampled  $\pi$ . Finally, structures and parameters  $\tilde{E}$  and  $\tilde{\theta}$  are sampled from their posterior distribution given  $X, Z$ .

## ■ 2.5 Summary

In this chapter, we introduced and developed many concepts central to the problem of studying time-varying interaction structures among a set of signals. First, the graphical model representation of joint probability distributions was discussed, both in the directed and undirected forms. Next, techniques for performing inference in graphical models and general joint probability distributions were discussed. Specifically, we discussed belief propagation for the problem of marginalization, various MCMC methods for sampling, and conjugate priors as a means to analytically evaluate posterior distributions.

The remainder of the chapter dealt with the switching state-space temporal interaction model, or SSIM. First, we described the concepts of dynamics Bayesian networks and linear Gaussian state space interaction models in order to motivate development of the SSIM graphical model. Next, we described the SSIM graphical model, which consists of a discrete switching sequence  $Z$ , a latent state sequence  $X$ , and an observed data sequence  $Y$ , linked together by several model parameters. Finally, we detailed a Gibbs sampling approach for obtaining samples from the SSIM joint distribution. In the next chapter, we will extend SSIM to allow for online inference, without specification of the number of transition models.



# Online Nonparametric Switching Temporal Interaction Model

In Chapter 2, we detailed the switching state-space temporal interaction model (SSIM) of Dzunic and Fisher, which described the dynamics of a linear Gaussian graphical model whose transition dynamics vary over time. Specifically, the model assumes an observed data sequence  $Y$ , a latent data sequence  $X$  whose evolution is governed by the transition dynamics, and a discrete sequence  $Z$  which indexes the transition model operating at any time. To perform inference on this graphical model, Dzunic and Fisher infer the latent sequence and switching states by use of a Gibbs sampler. We will refer to their algorithm for inference on SSIM herein as  $A_1$ .

In this chapter, we describe the development of an online nonparametric switching temporal interaction model (ONSTIM) inference algorithm, i.e. one that is able to incorporate observations as they arrive and that does not require specification of the number of transition model states. First, we motivate in greater detail the development of such an inference algorithm. Next, we describe the model that we assume for the generation of state sequences with an arbitrary number of states. Then, we give a high level overview of ONSTIM and provide justification for some of our design choices. ONSTIM consists of several subcomponents, including a run of  $A_1$ , one of two initialization procedures, and a run of a Gibbs sampler similar to  $A_1$  which we call  $A_2$ . The remainder of the chapter is devoted to describing in detail the initialization procedures and  $A_2$ .

### ■ 3.1 Motivation

Suppose we have taken observations of a time series  $Y_t$  from times  $t = 0$  through  $t = T$ . One can use the SSIM inference algorithm (see Algorithm 4) to obtain samples from the joint distribution of all variables in the SSIM graphical model, namely  $X$ ,  $Z$ ,  $\pi$ ,  $\tilde{E}$ ,  $\tilde{\theta}$ , and  $\xi$ , at all times from 0 through  $T$ . Suppose now that we obtain a new observation for time  $T + 1$  and we wish to incorporate this new information into our existing samples. To do so involves, at the very least, obtaining samples at  $T + 1$  of the latent data sequence,  $X_{T+1}$ , and of the switching state index sequence,  $Z_{T+1}$ . Taken further, one could attempt approaches that involve conditioning the structure and parameters on the new data, resampling the discrete state transition model, resampling the noise model, or conditioning samples of  $X$  and  $Z$  at older times on the new information.

If we are equipped only with the SSIM inference algorithm, however, the only option we have to incorporate the obseravtion  $Y_{T+1}$  into our inference procedure is to perform full smoothing on all times from  $t = 0$  through  $t = T + 1$ . In a variety of applications, it is desirable to employ faster methods to perform inference incorporating the new observation data without undertaking expensive computations over all of the previously received data. *Full* incorporation of  $Y_{T+1}$  into the previously computed values, i.e. conditioning all previously taken samples of all variables on  $Y_{T+1}$ , would effectively amount to full smoothing by running SSIM inference over all times. However, if we are only interested in performing inference at time  $T + 1$ , we can employ a much quicker filtering approach instead. If we wish to perform inference on time  $T + 1$  and also on times just before  $T + 1$ , we can perform fixed-lag smoothing, where the time horizon of interest determines the size of the lag. In this chapter. we will construct an online inference procedure that incorporates the new data  $Y_{T+1}$  by sampling variables at time  $T + 1$  and possibly in the *recent* past conditioned on the new observation.

Additionally, SSIM inference requires a priori specification of the number of different transition models,  $\{\tilde{E}, \tilde{\theta}\}_{1:K}$ . The number of different structure/parameter states present in an observed sequence is often unknown a priori, and specifying a number too low can at the very least force different states to merge, while choosing a number too high can cause splitting of a single state. It is thus also desirable to develop an inference procedure which not only allows for online inference, but in which the number of states  $K$  can also vary given the observed data. In the context of online inference,

this takes the form of modeling the arrival of new structure/parameter states over time with the arrival of new observations. To allow for this variation in model complexity, we will employ a Bayesian nonparametric approach to modeling the arrival of the new states, on which we provide some background information in the next section.

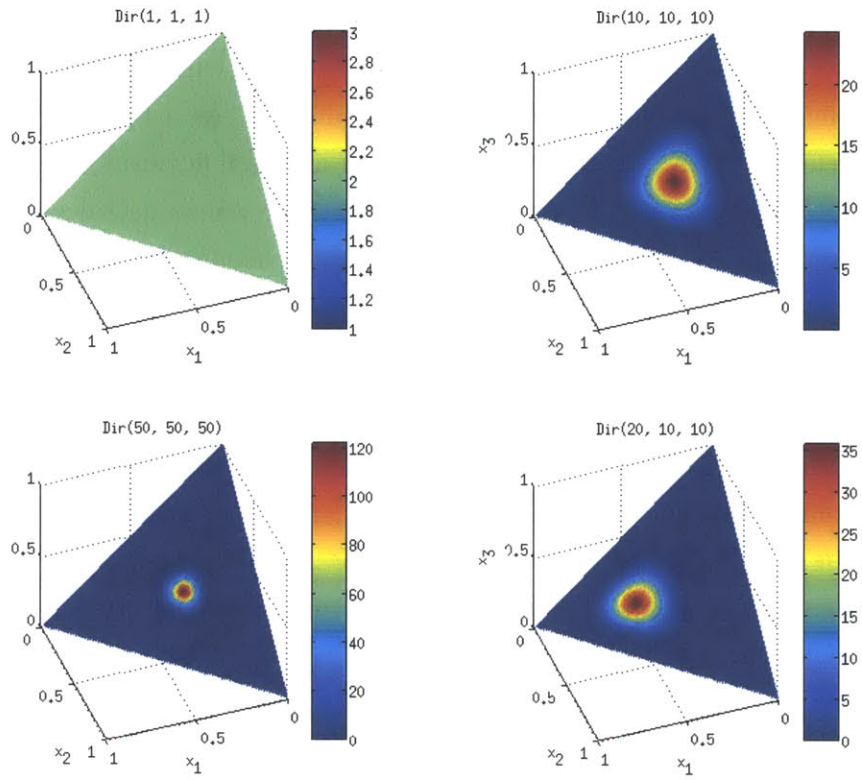
Given the application in this thesis to financial datasets, the motivation behind the development of an online inference scheme and a Bayesian nonparametric approach to the number of states is particularly apparent. The arrival of a new structure/parameter state could correspond to some sort of market regime shift in which dependencies among a set of financial instruments are shuffled and/or altered. As a trader or investor, it would be preferable to recognize this shift as soon as possible (ideally when the data arrives), and to incorporate knowledge of the shift into any subsequent decision making. In the next section, we will provide an overview of a generative model underlying the inference procedure of ONSTIM.

### ■ 3.2 State Sequence Generative Model

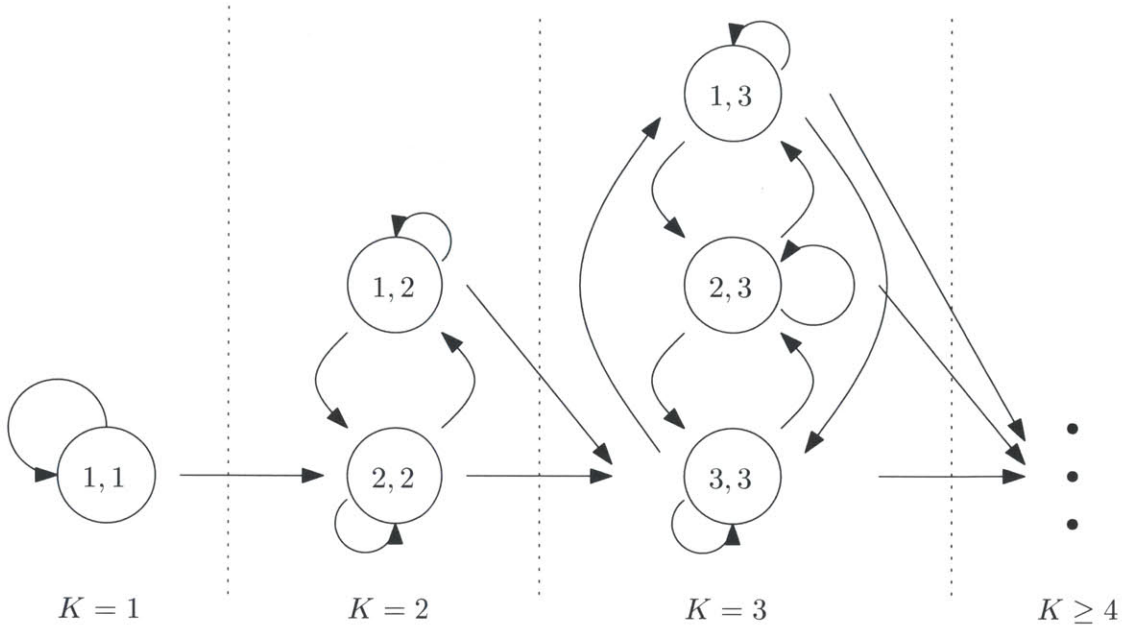
In Chapter 2, we described the graphical model and corresponding inference algorithm for the switching state-space interaction model, or SSIM. Inference on SSIM is parameterized by  $K$ , the total number of switching states present in the model, and this parameter must be assumed a priori. We provide an alternative parameterization of the model that lends itself to more effective use by users that have a sense of the prior on the arrival of new interaction structures and on the recurrence of the currently active interaction structure. To this end, we will describe a generative model with a parameter that characterizes the arrival of new states, and another that accounts for the recurrency of existing states, which we call  $\alpha_{new}$  and  $\alpha_{self}$  respectively.

In this section, the discrete state transition model  $\pi$  for the state sequence is fixed according to a set of user-chosen parameters instead of being sampled from a Dirichlet prior with hyperparameter  $\alpha$ . Note that this is approximately equivalent to placing an extremely strong prior on the transition model by means of very high values for entries of the Dirichlet hyperparameter  $\alpha$ , causing the Dirichlet prior to concentrate very strongly around its peak and effectively resemble a delta function around the  $\pi$  of choice, as shown in Figure 3.1. Fixing the transition model  $\pi$  manually allows us to directly study the effects of the choice of  $\pi$  on the results of inference.





**Figure 3.1.** Dirichlet distributions on the two-dimensional simplex, which represents the set of possible distributions on three discrete elements. Note that increasing the scale of  $\alpha$  without changing the relative magnitudes of the entries simply concentrates the distribution, while changing the relative magnitudes of the entries of  $\alpha$  moves the distribution around the simplex.



**Figure 3.2.** State diagram of the nested two-level Markov generative model, with each state indexed as  $(Z_t, K_t)$ . The outer chain is depicted horizontally, corresponding to transitions between the number of total states. The inner chain is depicted vertically, corresponding to transitions between states that have been instantiated.

**Markov Chain Model**

One approach to visualize the generative model described above is to interpret it as a nested two-level Markov chain, as shown in Figure 3.2. The outer chain is a transition model among  $K_t$ , the *number* of total instantiated states, while the inner chain, represented by  $\pi_K$ , is a transition model among the  $K$  instantiated states themselves. Note that since a different  $\pi_K$  holds for each value of  $K$ , it is useful to characterize the entire distribution with parameters from which both the outer Markov chain and all the inner transition models among the states can be computed. We achieve this by use of the abovementioned parameters,  $\alpha_{self}$  and  $\alpha_{new}$ , which characterize the probability of self-transition and new state instantiation respectively. The transition probabilities from any state to any other existing state are chosen to be uniform, so as to simplify the model for ease of reasoning over recurrence and new state arrival probabilities.

Let  $Z_t$  denote the active model index at time  $t$ , and let  $K_t$  denote the total number of states that have been instantiated up to time  $t$ . We can express the distribution of

$Z_{t+1}$  given  $Z_t$  and  $K_t$  as:

$$P(Z_{t+1}, K_{t+1} | Z_t, K_t) = \begin{cases} \frac{\alpha_{self}}{\alpha_{self} + \alpha_{new} + K_t - 1} & \text{if } Z_{t+1} = Z_t, K_{t+1} = K_t, \\ \frac{1}{\alpha_{self} + \alpha_{new} + K_t - 1} & \text{if } Z_{t+1} \neq Z_t, K_{t+1} = K_t, \\ \frac{\alpha_{new}}{\alpha_{self} + \alpha_{new} + K_t - 1} & \text{if } Z_{t+1} = K_{t+1} = K_t + 1. \end{cases} \quad (3.1)$$

The conditional distribution described in Equation 3.1 offers three possibilities: the system can either return to its original state, transition to a different existing state, or instantiate a new state. The value of  $\alpha_{self}$  is typically chosen to be very high, implying a very strong prior on recurrence to the current state. The value of  $\alpha_{new}$ , though typically not nearly as high as  $\alpha_{self}$ , is still chosen to be significantly higher than 1. Thus, for low  $K_t$ , specifically for  $K_t < \alpha_{new}$  the probability of instantiating a new state is higher than the *sum* of the probabilities of transitioning to any existing state besides the current state.

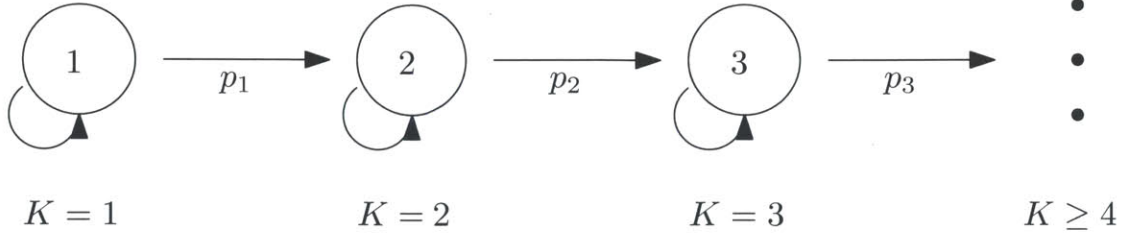
So far, we have described the full state transition process among joint states of the form  $(Z_t, K_t)$ . Since we are interested in the distribution over the total number of states at time  $t$ , we will consider the transitions over time between macrostates which we define by the total number of instantiated states up to a given time, denoted by  $K_t$ . Obtaining the transition distribution between the macrostates  $K_t$  is equivalent to marginalizing out the state variable  $Z_t$ . This gives us the conditional distribution:

$$P(K_{t+1} | K_t) = \begin{cases} \frac{\alpha_{self} + K_t - 1}{\alpha_{self} + \alpha_{new} + K_t - 1} & \text{if } K_{t+1} = K_t, \\ \frac{\alpha_{new}}{\alpha_{self} + \alpha_{new} + K_t - 1} & \text{if } K_{t+1} = K_t + 1. \end{cases} \quad (3.2)$$

The corresponding Markov chain diagram for this transition distribution is shown in Figure 3.3. As more states are instantiated, the denominator for the probability of new state instantiation increases while the numerator remains constant, implying a lower rate of new state instantiation over time.

The generative model described above for the full random variable  $(Z_t, K_t)$  has many similarities to the sticky hierarchical Dirichlet process hidden Markov model (HDP-HMM) of Fox et al., for whose details we refer the reader to [2]. Both models describe the set of possible switching states and their transition kernel nonparametrically. However, while state in the sticky HDP-HMM is indexed only by  $Z_t$ , in our generative model, state is indexed by the joint variable  $(Z_t, K_t)$ , and transitions do not exist between all possible pairs of  $(Z_t, K_t)$ .





**Figure 3.3.** Markov chain macrostate diagram, with macrostates corresponding to total number of instantiated states. Transitions between states are labelled with probabilities, as defined in Equation 3.2.

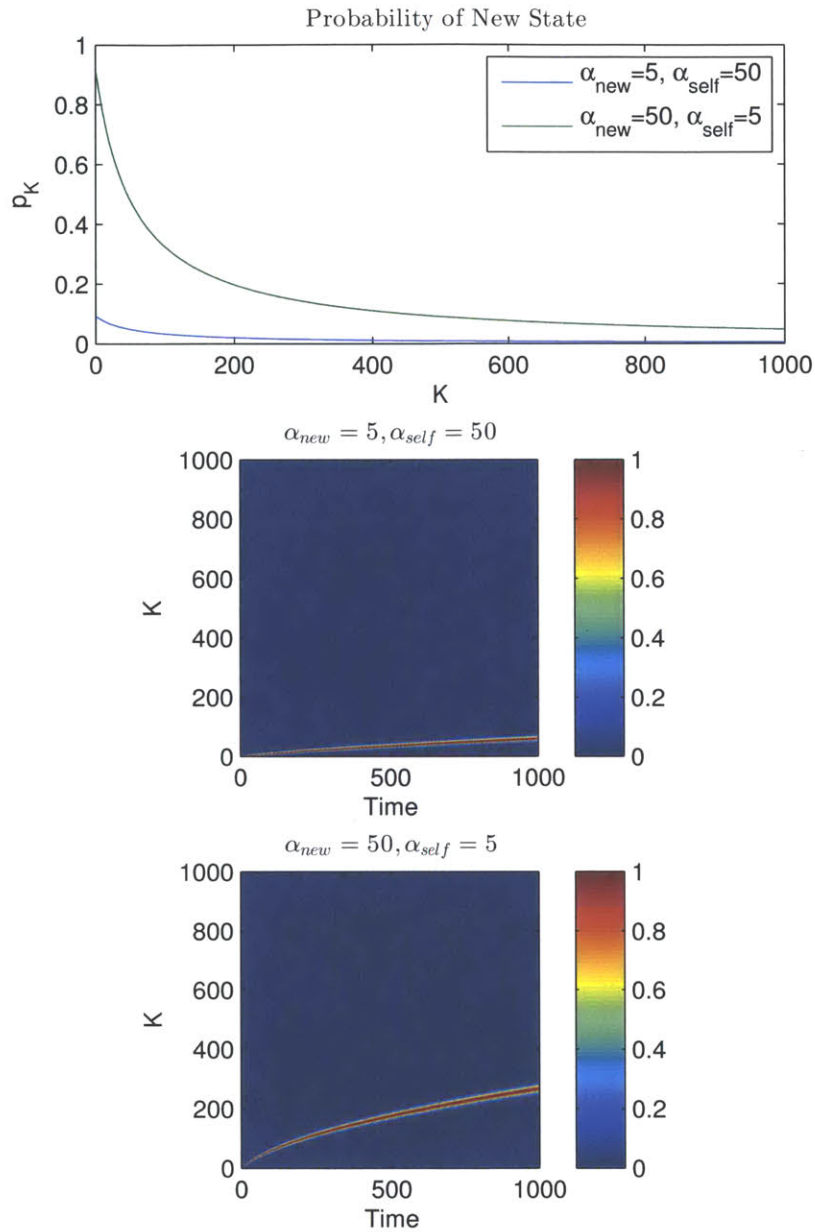
### Distribution of $K_t$

A quantity of interest is the distribution over the number of instantiated states at time  $t$ , namely  $K_t$ , as a function of  $t$  and the parameters  $\alpha_{self}$  and  $\alpha_{new}$ . Denote by  $p_K$  the probability of instantiating a new state if  $K$  states currently exist, which corresponds to the second case in Equation 3.2. Although the generative model can instantiate an arbitrary number of states, note that at time  $t$ , it is not possible for more than  $t$  states to have been instantiated, i.e.  $K_t \leq t$ . Thus, for any finite  $t$ , we can reason over the distribution of  $K_t$  in terms of the distribution of  $K_{t-1}$  and  $p_{K_{t-1}}$ . Specifically, if we view the distribution of  $K_t$  as a  $t$ -dimensional vector, where the  $k^{\text{th}}$  entry represents  $P(K_t = k)$ , we can write the following recurrence:

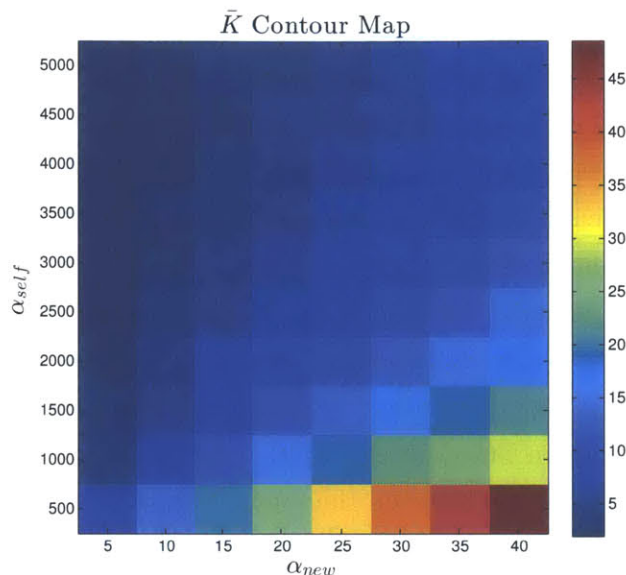
$$P(K_t = k) = \begin{cases} (1 - p_k)P(K_{t-1} = k) & \text{if } k = 1, \\ p_{k-1}P(K_{t-1} = k-1) + (1 - p_k)P(K_{t-1} = k) & \text{if } 1 < k < t, \\ p_{k-1}P(K_{t-1} = k-1) & \text{if } k = t. \end{cases} \quad (3.3)$$

If we wish to compute the distribution of  $K_t$  for  $1 \leq t \leq T$ , we must compute  $P(K_t = k)$  for  $1 \leq k \leq t$ , and we must do this for every  $t$  between 1 and  $T$ . Computing the distribution of  $K_t$  is thus an  $O(T^2)$  procedure.

While  $K_t$  is positive for all  $k \leq t$ , the vast majority of the possible values of  $K_t$  occur with very low probability. Instead of attempting to visualize the full distribution, we will instead display summary statistics, namely the expected value and mode, of the distribution of  $K_t$  as a function of  $t$  for certain parameter values of  $\alpha_{new}$  and  $\alpha_{self}$ . In Figure 3.4, we display the probability of new state instantiation and the distribution of  $K_t$  for  $(\alpha_{new}, \alpha_{self}) = (50, 5)$  and  $(5, 50)$ , where we set the maximum time to be  $T = 1000$ .



**Figure 3.4.** The top pane shows probability of new state instantiation for  $(\alpha_{new}, \alpha_{self}) = (50, 5)$  and  $(5, 50)$  as a function of the number of instantiated states  $K$ . As  $K$  increases,  $\alpha_{new}$  decreases relative to the sum of the transition parameters for all other states, resulting in decreasing  $p_K$ . The middle and bottom panes depict the distribution of  $K_t$ , the total number of instantiated states at time  $t$ , as a function of  $\alpha_{new}$  and  $\alpha_{self}$ . For the purpose of readability, each column of the middle and bottom panes has been normalized so that its maximum value is 1.



**Figure 3.5.** Contour map of the average number of instantiated states across samples ( $\bar{K}$ ) in the  $(\alpha_{new}, \alpha_{self})$ -plane for  $T = 1000$  time points. Data points were computed with 2,000 samples each per  $(\alpha_{new}, \alpha_{self})$  pair.

Note that high values of  $\alpha_{self}$  encourage the process to remain in the current state, thereby suppressing the instantiation of new states, while high values of  $\alpha_{new}$  directly encourage instantiation of new states. It is thus of interest to study the interaction between  $\alpha_{self}$  and  $\alpha_{new}$ . Shown in Figure 3.5 are the average number of instantiated states in simulations run with different values of  $\alpha_{self}$  and  $\alpha_{new}$ . The highest values of  $\bar{K}$  occur in the bottom right, where  $\alpha_{new}$  is high and  $\alpha_{self}$  low, while the lowest values of  $\bar{K}$  occur in the top left.

### ■ 3.3 Overview of ONSTIM Inference

So far in this chapter, we have described the relevant generative model that underlies our inference procedure. We now switch gears to describing the procedure of inference itself. ONSTIM allows for an online approach to detecting interaction structures by dividing the observed sequence into many batches, each of length  $B$ , and only performing inference on the most recent batch. We now detail the batch framework employed in ONSTIM, and then describe how inference is performed using this framework. The procedure is concisely documented in Algorithm 3.3, and a visualization of the batch

framework is presented in Figure 3.3.

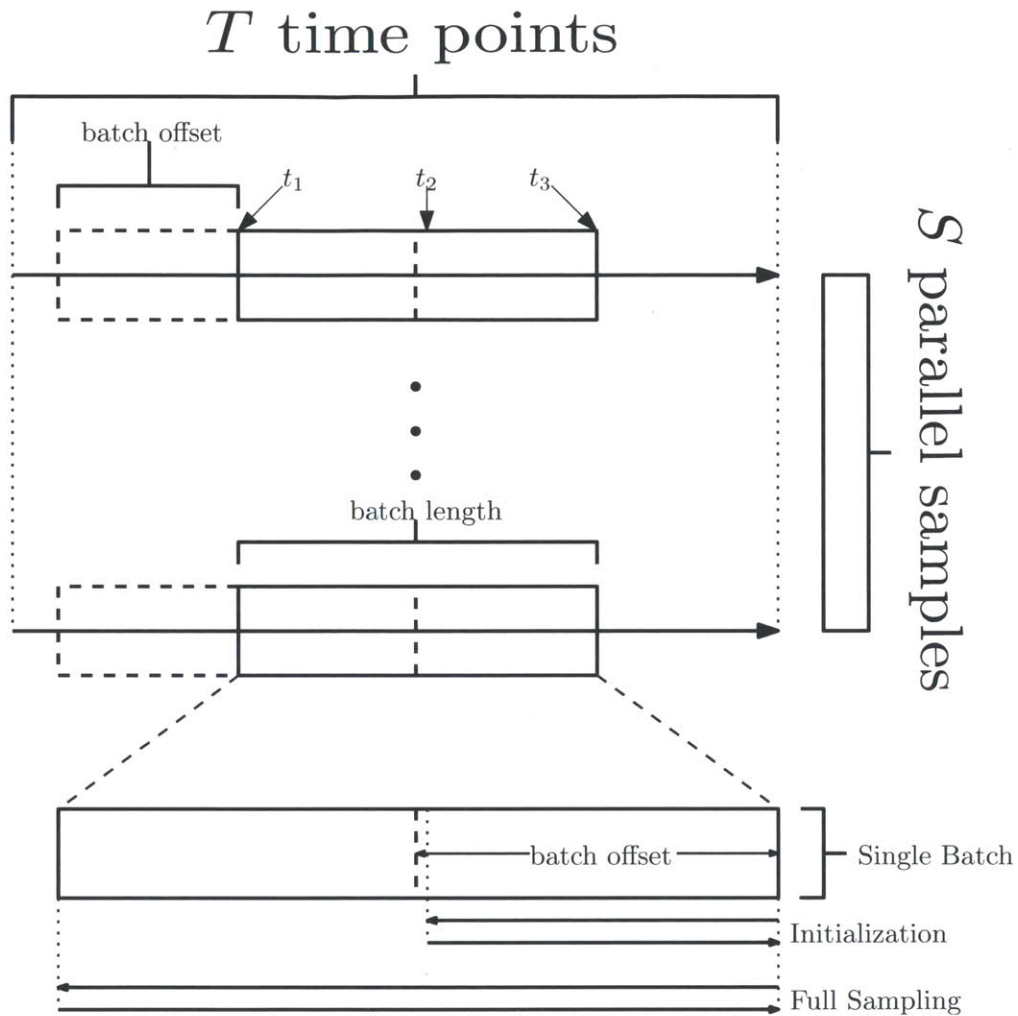
The batch framework for any experiment consists of the subdivision of all time points in the observed sequence into possibly overlapping sets called batches. A batch framework is parameterized by the batch length, which we denote here as  $B$ , and by the batch offset. The batch offset is the distance in time from the last point of a certain batch to the last time point of the *subsequent* batch. Since batch length and batch offset are constant over the course of an experiment, the batch offset can also be interpreted as the number of points in a given batch that are not members of the subsequent batch. For instance, if batch 1 contains times  $t = 1$  through 10 and the batch offset is set to 2, then batch 2 will contain time points  $t = 3$  through 12. We will explore the effect of the choice of the batch offset scheme on the inference results in Chapter 4.

Recall from Chapter 2 that averaging over many samples provides an unbiased estimator of expected value that has much less variance than a single sample. Therefore, in order to better model the full joint distribution over all variables of interest, instead of sampling only one sequence of batches, we sample  $S$  parallel batch sequences, as seen in Figure 3.3. Equipped with multiple samples, we can better characterize the expected value of any function of the joint distribution that is of interest to us. For example, if we are interested in the probability of some event  $A$  occurring, we can use the approximation  $p(A) \approx \frac{1}{S} \sum_{i=1}^S \mathbb{1}_{A,i}$ .

We now provide a brief high-level overview of ONSTIM. In each of the  $S$  parallel batch sequences, inference on the first batch is performed by directly using the SSIM inference algorithm. After a sufficient burn-in period, a single sample is taken from the sample chain, which is then used to help initialize the next batch. Subsequent batches are initialized partially with samples from previous batches, and partially with samples obtained from initialization procedures described later in the chapter.

Suppose that all time points from  $t_1$  through  $t_2 - 1$  are shared between two consecutive batches. In the latter batch, those shared time points are directly initialized to their sampled values from the previous batch. Time points that are *new* to the latter batch are initialized using either Algorithm 8 or Algorithm 9. If a new interaction structure and transition parameter model are sampled during this initialization, the model is instantiated into the set of available transition models. Regardless of whether a new model is instantiated, Gibbs sampling is then performed for a fixed number of rounds using Algorithm 10, and the final sample is taken and stored.





**Figure 3.6.**  $S$  samples are taken in parallel. Each sample is divided into batches of length  $B$ , which move forwards by the batch offset. Gibbs sampling is performed on each batch. The times  $t_1$ ,  $t_2$ , and  $t_3$  correspond to their respective descriptions in Algorithm 3.3.  $t_1$  is the first time point of the current batch, shown as a solid block,  $t_3$  is the final time point of the current batch, and  $t_2$  is the first time point to come directly *after* the last time point in the previous batch, which is shown as a dashed block. Note that  $t_2$  was not in the previous batch. A magnified version of a batch from the final sample is shown at the bottom of the figure, with the batch offset, regime for initialization, and regime for full sampling all labelled.

The batch framework allows for simple filtering and fixed-lag smoothing approaches to online inference. Filtering, the process of performing inference with data only through the present, can be accomplished at some time  $t$  by considering samples taken from the batch which ends at time  $t$ . Fixed-lag smoothing, on the other hand, allows for the utilization of some fixed horizon of data after the time point of interest to improve inference results. Fixed-lag smoothing for some lag  $\lambda$  can be similarly performed by using samples taken from the batch ending at  $t + \lambda$ , as this incorporates information from  $\lambda$  time points into the future. Note that this implies that fixed-lag smoothing cannot be achieved for  $\lambda \geq B$ , as there exists no batch which contains samples from times further apart than  $B - 1$ .

Apart from allowing for online inference, ONSTIM allows for nonparametric modelling of the number of structure/parameter states as the inference procedure moves through time. In each new batch, there is a positive probability of arrival of a new structure/parameter state, which is sampled from the prior on structure and parameters. The probability of that the system enters a new state at any given time is dependent both on the prior probability of the arrival of new states and also on the likelihood that the system entered a new state given the observed data.

```

Data:  $Y, S, K, B, \text{offset}$ 
Result:  $X_{1:S,1:\text{length}(Y)}, Z_{1:S,1:\text{length}(Y)}, \{\tilde{E}, \tilde{\theta}\}_{1:S,1:K_s}$ 
 $n = 1 + \frac{\text{length}(Y) - B}{\text{offset}};$ 
for  $s = 1 : S$  do
   $X_{s,0:B}, Z_{s,1:B}, \{\tilde{E}, \tilde{\theta}\}_{s,1:K} = A_1(Y_{0:B}, K);$ 
   $K_s = K;$ 
  for  $b = 2 : n$  do
     $t_1 = (b - 1) \cdot \text{offset} + 1;$  // start of new batch
     $t_2 = B + (b - 2) \cdot \text{offset} + 1;$  // time point after end of previous batch
     $t_3 = B + b \cdot \text{offset};$  // end of new batch
     $X_{s,t_2:t_3}, Z_{s,t_2:t_3} = \text{Initialize}(t_2, t_3, Y_{s,t_2:t_3}, X_{s,t_2-1}, Z_{s,t_2-1}, \{\tilde{E}, \tilde{\theta}\}_{s,1:K_s});$ 
    if  $K_s + 1 \in Z_{s,t_2:t_3}$  then
      Instantiate  $\{\tilde{E}, \tilde{\theta}\}_{s,K_s+1} \sim P(\{\tilde{E}, \tilde{\theta}\} | \{X_{s,t}, X_{s,t-1}\}_{t:Z_{s,t=K_s+1}});$ 
       $K_s = K_s + 1;$ 
       $X_{s,t_1:t_3}, Z_{s,t_1:t_3}, \{\tilde{E}, \tilde{\theta}\}_{s,1:K_s} = A_2(Y_{s,t_1:t_3}, X_{s,t_1:t_3}, Z_{s,t_1:t_3}, \{\tilde{E}, \tilde{\theta}\}_{s,1:K_s});$ 
    end
  end
end

```

**Algorithm 7.** High-level overview of ONSTIM, with  $A_1, A_2$ , and *Initialize* called as subroutines.  $A_1$  is the full Gibbs sampling procedure of Dzunic and Fisher [1], described in Chapter 2.  $A_2$  is the post-initialization Gibbs sampling procedure we employ, described in Section 3.7. *Initialize* is the initialization routine we employ, which can either be Approach 1, described in Section 3.5, or Approach 2, described in Section 3.6. Instantiating a new state is performed in the same manner as in  $A_1$ .

### ■ 3.4 Complexity of Exact Inference

As described in Section 3.3, after initialization of  $X$  and  $Z$  variables in a new batch, we immediately choose whether or not to instantiate a new structure/parameter state. If we decide to instantiate a new state, a new structure/parameter state is sampled from the posterior conditioned on the pairs of latent data sequence points  $(X_t, X_{t-1})$  for times  $t$  such that  $Z_t = K + 1$ . Otherwise, no new state is instantiated. In either case, after initialization, Gibbs sampling proceeds almost identically to Gibbs sampling in SSIM, with either  $K$  or  $K + 1$  transition models available. Note that this approach implies that for a new state to be instantiated during sampling for a batch, the instantiation *must* occur during initialization, with no opportunity for sampling a new state during

the rounds of Gibbs sampling.

It may seem preferable to instead perform the sampling of  $X$  and  $Z$  in the Gibbs sampling rounds taking the  $K + 1^{\text{st}}$  state to represent the prior distribution over structure and parameters instead of any specific structure/parameter pair, thereby delaying specializing the new state to a particular transition model until desirable. Unfortunately, due to issues of computational tractability, running such a sampling procedure is not possible. Although it is possible to sample  $Z$  while accounting for the  $K + 1^{\text{st}}$  state as the prior distribution, it is difficult to sample  $X$  similarly. In this section, we will show why this is the case, and we will discuss approximate solutions that attempt to tackle this issue.

### ■ 3.4.1 Intractable Message Passing for $X$

In SSIM inference, recall that one of the sampling steps of the Gibbs sampler is that of sampling the latent data sequence  $X \sim P(X|Z, Y, \tilde{E}, \tilde{\theta})$ . This is accomplished by a backwards message passing step, followed by a round of forwards sampling. The backwards messages are given by

$$m^t(X_t) = \int_{X_{t+1}} P(X_{t+1}|X_t, Z_{t+1})P(Y_{t+1}|X_{t+1})m^{t+1}(X_{t+1})dX_{t+1}. \quad (3.4)$$

Samples of  $X_t$  conditioned on  $X_{0:t-1}, Z, Y, \tilde{E}$ , and  $\tilde{\theta}$  are then obtained by sampling

$$X_t \sim P(X_t|X_{0:t-1}, Z, Y, \tilde{E}, \tilde{\theta}) \propto P(X_t|X_{t-1}, Z_t)P(Y_t|X_t)m^t(X_t). \quad (3.5)$$

In SSIM inference,  $Z_t$  takes on values in  $\{1, \dots, K\}$ , where each possible value indexes a linear Gaussian transition model.

If  $Z_t = K + 1$ , then we have

$$m^t(x) = \int_{X_{t+1}} P(X_{t+1}|x; \beta, \gamma)P(Y_{t+1}|X_{t+1})m^{t+1}(X_{t+1})dX_{t+1}. \quad (3.6)$$

Let us focus on the term  $P(X_{t+1}|X_t; \beta, \gamma)$ . Note first of all that we can decompose this probability into a product across the entries of the vector  $X_{t+1}$ , since we assume independence across the probability distributions of each element of the vector. Thus,

$$P(X_{t+1}|X_t; \beta, \gamma) = \prod_{i=1}^N P(X_{t+1}^i|X_t; \beta, \gamma). \quad (3.7)$$



Next, note that  $P(X_{t+1}^i|X_t; \beta, \gamma)$  is a probability taken over the prior over structures and parameters by marginalizing out all parent sets and parameters. We can write out the marginalization of parent sets explicitly as

$$\prod_{i=1}^N P(X_{t+1}^i|X_t; \beta, \gamma) = \prod_{i=1}^N \sum_{\tilde{pa}(i)} P(X_{t+1}^i|X_t^{\tilde{pa}(i)}; \gamma) P(\tilde{pa}(i); \beta). \quad (3.8)$$

The term  $P(X_{t+1}^i|X_t^{\tilde{pa}(i)}; \gamma)$  is distributed according to a Student's  $t$ -distribution while  $P(\tilde{pa}(i); \beta)$  is a constant determined by the prior on structure. Thus, the weighted sum  $\sum_{\tilde{pa}(i)} P(X_{t+1}^i|X_t^{\tilde{pa}(i)}; \gamma) P(\tilde{pa}(i); \beta)$  takes the form of a mixture of  $t$ -distributions, analogous to a Gaussian mixture model.  $P(X_{t+1}|X_t; \beta, \gamma)$  is then distributed according to a product of  $t$ -mixture models.

Consider the case of  $t = T - 1$ , in which  $m^{t+1}(X_{t+1}) = m^T(X_T) = 1 \forall X_T$ . The observation model characterized by the term  $P(Y_T|X_T)$  is multivariate Gaussian, so if  $Z_T = K + 1$ , computing the message

$$m^{T-1}(X_{T-1}) = \int_{X_T} P(X_T|X_{T-1}; \beta, \gamma) P(Y_T|X_T) dX_T \quad (3.9)$$

requires integrating over the product of a Gaussian and a product of  $t$ -mixture models, which has no closed form analytical solution.

### ■ 3.4.2 Alternative Approaches

Since computation of the message described in the above equation is intractable, it is natural to attempt to approximate the integral with something that is analytically tractable. In this section, we discuss potential approaches for approximating the computation of the message with techniques based on Gaussian distributions. Although we do not implement any of these approximations in this thesis, they are potential avenues for further exploration.

First, we propose a naïve approach, namely that of approximating  $P(X_{t+1}|X_t; \beta, \gamma)$  with a Gaussian distribution. There are several different ways to perform this approximation. A simple solution is to replace the distribution  $P(X_{t+1}|X_t; \beta, \gamma)$  with a Gaussian possessing the same mean and covariance. This method has the advantage of being very easy to compute and implement, since it fits directly into the Gaussian message passing framework. However, it may be a gross oversimplification of the original distribution.

Recall that  $P(X_{t+1}|X_t; \beta, \gamma)$  is a product of  $t$ -mixture models. We will attempt to qualitatively characterize this distribution and compare it to a Gaussian approximation. If  $X$  is a  $N$ -dimensional vector, our constraints allow each node in  $X$  to have up to  $M$  parents, and each node is required to have itself as a parent, then each node can have  $\sum_{i=0}^{M-1} \binom{N-1}{i}$  possible parent sets. Thus, each  $t$ -mixture model corresponding to  $\sum_{\tilde{pa}(i)} P(X_{t+1}^i | X_t^{\tilde{pa}(i)}; \gamma) P(\tilde{pa}(i); \beta)$  for some node  $i$  has up to  $\sum_{i=0}^{M-1} \binom{N-1}{i}$  modes. Since each  $t$ -mixture model is one of  $N$  factors of the overall distribution, the distribution of  $P(X_{t+1}|X_t; \beta, \gamma)$  has up to  $(\sum_{i=0}^{M-1} \binom{N-1}{i})^N$  modes, significantly more than the unimodal Gaussian distribution. Constructing a Gaussian distribution with the same mean and covariance as  $P(X_{t+1}|X_t; \beta, \gamma)$  could yield significant density in regions of the support where the original distribution has little density and vice versa. If the modes corresponding to different parent set choices happen to be very close to each other, then the unimodal Gaussian approximation may be a reasonable technique.

Another Gaussian-based approach is to replace the target distribution  $P(X_{t+1}|X_t; \beta, \gamma)$  with a Gaussian mixture model instead of just a single Gaussian. This could be achieved by replacing each individual  $t$ -distribution with a Gaussian distribution of equal mean and covariance. Then, Gaussian mixture models could be constructed analogously to the  $t$ -mixture models, and the approximating distribution could be written as the product of Gaussian mixture models. While this approximation is much more faithful to the original distribution, performing message passing with Gaussian mixture models is extremely computationally expensive, but not technically analytically intractable. Since the product of Gaussian mixture model with  $a$  components and a Gaussian mixture model with  $b$  components yields a Gaussian mixture model with  $ab$  components, running message passing with Gaussian mixture model messages leads to an exponential blowup in the size of the parameterization of the mixture model. While this directly leads to computational intractability of the approach, further heuristic techniques such as merging or truncating the lowest weight components have been employed in the past [9].

### ■ 3.5 Batch Initialization of $X$ and $Z$ : Approach 1

The first step in the inference procedure for a given batch of data is initialization of the discrete states, the  $Z$  variables, and the latent data sequence, the  $X$  variables.

Dzunic et al. also perform an initialization of the Gibbs sampler before proceeding to actual sampling, but our first approach to batch initialization is significantly different from theirs. First of all, we can take advantage of samples of  $X$  and  $Z$  taken from the previous batch. Second, we wish to propose the possibility of the arrival of a new structure and parameter state drawn from the prior over interaction structures and parameters.

Let us define  $t_1, t_2$ , and  $t_3$  as in Algorithm 3.3. Suppose the batch that we wish to initialize runs from time  $t_1$  through  $t_3$ . Given our allocation of time points to batches, this implies that state and sequence values at times  $t_1$  through  $t_2$  were sampled already in the previous batch. We will initialize  $X_i$  and  $Z_i$  to their respective values from the previous batch for  $i \in [t_1, t_2 - 1]$ .

Next, we must develop a method to initialize  $X_i$  and  $Z_i$  for  $i \in [t_2, t_3]$ . We will employ a greedy initialization approach, in which observations for times greater than  $i$  are ignored in the initialization of  $X_i$  and  $Z_i$ . The initialization procedure iterates over the times in the outer loop, and between sampling  $Z_i$  and  $X_i$  in the inner loop.

**Data:**  $t_2, t_3, Y, X_{t_2-1}, Z_{t_2-1}, \{\tilde{E}, \tilde{\theta}\}_{1:K}$   
**Result:**  $X_{t_2:t_3}, Z_{t_2:t_3}$   
**7 for**  $i = t_2 : t_3$  **do**  
    | Sample  $Z_i \sim P(Z_i = z | Z_{i-1}, Y_i, X_{i-1}), z \in \{1, \dots, K + 1\}$   
    | Sample  $X_i \sim P(X_i | X_{i-1}, Y_i, E_{Z_i}, \Theta_{Z_i}; \beta, \gamma)$   
**end**

**Algorithm 8.** High-level batch initialization of  $X$  and  $Z$ : Approach 1. We use the  $t_2, t_3$  notation for the sake of consistency with Algorithm 3.3. Details of the sampling steps are provided below.

### ■ 3.5.1 Initialization of $Z_i \sim P(Z_i | Z_{i-1}, Y_i, X_{i-1})$

The actual process of sampling from the conditional distributions of  $Z_i$  and  $X_i$  listed above is complicated by modeling the arrival of a new state that is sampled from the prior over structures and parameters. Conceptually, if a transition from time  $i$  to  $i + 1$  is better modelled by the prior across all structures and parameters than by any of the existing  $K$  structure and parameter states, we wish to allow a proposal of a new state sampled from the prior conditional on such transitions. Unfortunately, while computing the conditional probability distribution of a transition  $P(X_i | X_{i-1}; \beta, \gamma)$  is analytically

tractable, the resulting distribution is a matrix-T distribution instead of a multivariate Gaussian distribution. Consequently, no further useful analytical operations, such as multiplication by multivariate Gaussian valued random variables, necessary for any message passing algorithm, can be performed on the resulting distribution.

However, computing  $P(X_i|X_{i-1}; \beta, \gamma)$  is necessary for sampling both  $Z_i$  and  $X_i$ . Instead of

$$P(Z_i|Z_{i-1}, Y_i, X_{i-1}) \quad (3.10)$$

$$= \int_{X_i} P(Z_i, X_i|Z_{i-1}, Y_i, X_{i-1}) dX_i \quad (3.11)$$

$$= \int_{X_i} P(X_i|Z_i, Z_{i-1}, Y_i, X_{i-1}) P(Z_i|Z_{i-1}, Y_i, X_{i-1}) dX_i \quad (3.12)$$

$$= \int_{X_i} P(X_i|Z_i, Y_i, X_{i-1}) P(Z_i|Z_{i-1}) dX_i \quad (3.13)$$

$$\propto P(Z_i|Z_{i-1}) \int_{X_i} P(X_i, Y_i|Z_i, X_{i-1}) dX_i \quad (3.14)$$

$$= P(Z_i|Z_{i-1}) \int_{X_i} P(X_i|Z_i, X_{i-1}) P(Y_i|X_i) dX_i. \quad (3.15)$$

For  $Z_i \in \{1, \dots, K\}$ , the expression  $P(X_i|Z_i, X_{i-1})$  is a multivariate Gaussian, and therefore  $X_i$  can be integrated out analytically quite simply. If  $Z_i = K + 1$ , however, we are computing the conditional probability of  $X_i$  given  $X_{i-1}$  under the prior over structures and parameters, such that  $P(X_i|Z_i = K + 1, X_{i-1}) = P(X_i|X_{i-1}; \beta, \gamma)$ . Since this is not a multivariate Gaussian distribution, we must adopt a non-analytical approach to allow us to integrate over  $X_i$ . We will proceed by sampling  $N$  instantiations of structures and parameters from the prior over structures and parameters. Note that

$$P(X_i|X_{i-1}; \beta, \gamma) = \sum_E \int_{\Theta|E} P(X_i|X_{i-1}, E, \Theta) P(E, \Theta; \beta, \gamma) d\Theta \quad (3.16)$$

$$= \mathbb{E}_{E, \Theta} [P(X_i|X_{i-1}, E, \Theta)] \quad (3.17)$$

$$= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j) \quad (3.18)$$

$$\approx \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j). \quad (3.19)$$

Thus, for  $Z_i = K + 1$ , we have

$$P(Z_i|Z_{i-1}, Y_i, X_{i-1}) = P(Z_i|Z_{i-1}) \int_{X_i} P(X_i|Z_i, X_{i-1})P(Y_i|X_i)dX_i \quad (3.20)$$

$$\approx P(Z_i|Z_{i-1}) \int_{X_i} \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j)P(Y_i|X_i)dX_i \quad (3.21)$$

$$= \frac{P(Z_i|Z_{i-1})}{N} \sum_{i=j}^N \int_{X_i} P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j)P(Y_i|X_i)dX_i. \quad (3.22)$$

Thus, given  $Y_i$ ,  $Z_{i-1}$ , and  $X_{i-1}$ , we have constructed up to a normalization factor the complete probability distribution of  $Z_i$ , for values that corresponding to existing structure and parameter states, and also for the possibility of the arrival of a new state sampled from the prior over structures and parameters. We then normalize the distribution to sum to 1 and sample  $Z_i$ , whose value we store.

In conclusion, to initialize  $Z_i$ , we sample  $Z_i \sim P(Z_i|Z_{i-1}, Y_i, X_{i-1})$ , where

$$P(Z_i|Z_{i-1}, Y_i, X_{i-1}) \propto \begin{cases} P(Z_i|Z_{i-1}) \int_{X_i} P(X_i|X_{i-1}, E_{Z_i}, \Theta_{Z_i})P(Y_i|X_i)dX_i & \text{if } Z_i \in \{1, \dots, K\} \\ \frac{P(Z_i|Z_{i-1})}{N} \sum_{i=j}^N \int_{X_i} P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j)P(Y_i|X_i)dX_i & \text{if } Z_i = K + 1. \end{cases} \quad (3.23)$$

### ■ 3.5.2 Initialization of $X_i \sim P(X_i|X_{i-1}, Z_i, Y_i)$

The procedure for sampling  $X_i$  given  $X_{i-1}$ ,  $Z_i$ , and  $Y_i$ , is embedded into the procedure for sampling  $Z_i$  described above, but with  $X_i$  marginalized out. We have

$$P(X_i|X_{i-1}, Y_i, Z_i) \propto P(X_i, Y_i|X_{i-1}, Z_i) \quad (3.24)$$

$$= P(Y_i|X_i, X_{i-1}, Z_i)P(X_i|X_{i-1}, Z_i) \quad (3.25)$$

$$= P(Y_i|X_i)P(X_i|X_{i-1}, Z_i). \quad (3.26)$$

For  $Z_i \in \{1, \dots, K\}$ , this reduces to  $P(Y_i|X_i)P(X_i|X_{i-1}, E_{Z_i}, \Theta_{Z_i})$ . For  $Z_i = K + 1$ , we must resort to a sampling approach reflecting the approach taken to compute the conditional probability of  $Z_i = K + 1$  in the previous subsection.

We have shown already that

$$P(X_i|X_{i-1}; \beta, \gamma) \approx \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j). \quad (3.27)$$



Thus, for  $Z_i = K + 1$ ,

$$P(X_i|X_{i-1}, Y_i, Z_i) \propto \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j) P(Y_i|X_i). \quad (3.28)$$

In conclusion, to initialize  $X_i$ , we sample  $X_i \sim P(X_i|X_{i-1}, Z_i, Y_i)$ , where

$$P(X_i|X_{i-1}, Z_i, Y_i) \propto \begin{cases} P(X_i|X_{i-1}, E_{Z_i}, \Theta_{Z_i}) P(Y_i|X_i) & \text{if } Z_i \in \{1, \dots, K\} \\ \frac{1}{N} \sum_{j=1}^N P(X_i|X_{i-1}, \hat{E}_j, \hat{\Theta}_j) P(Y_i|X_i) & \text{if } Z_i = K + 1. \end{cases} \quad (3.29)$$

### ■ 3.6 Batch Initialization of $X$ and $Z$ : Approach 2

In the previous section, we described an approach to initializing the values of  $X$  and  $Z$  at new time points in a batch in which  $X_t$  and  $Z_t$  are alternately sampled as  $t$  is incremented from the first new time point in the batch to the last new point in the batch. However, that initialization method has two significant shortcomings. First of all, a sample of  $Z_t$  or  $X_t$  only incorporates observed sequence data up through  $t$ , omitting new data provided to the batch at times after  $t$ , thereby eschewing a potentially useful source of information. Second, computing the probability  $P(Z_t = K + 1)$  required marginalizing over  $X_t$  by taking some number of samples of structure and parameter states from the prior. Significant variance is introduced into the sampling procedure unless a large number of samples from the prior is taken, which requires significant time. It was mentioned in Section 3.4 that while performing message passing for  $X$  is intractable if the  $K + 1^{\text{st}}$  state represents the prior over all structures and parameters, performing message passing for  $Z$  is tractable. Since performing full message passing for  $Z$  requires an existing sample of  $X$ , the second approach will obtain a sample of  $X$  first and then sample  $Z$  conditioned on  $X$ .

In this section, we will describe a second approach for initialization of  $X$  and  $Z$  which closely mirrors that employed in  $A_1$ .  $X$  is sampled from  $Y$  using a simple Kalman filter with the transition matrix  $A$  assumed to be equal to  $I$  and with the transition covariance fixed to a constant input parameter  $\Sigma$ . Let us denote by  $t_2$  the time point directly after the last point of the previous batch, which is the earliest point that we must initialize. Instead of sampling  $X_{t_2}$  conditioned only on  $Y$  as in  $A_1$ , however, we now sample  $X_{t_2}$  conditioned on  $Y$  and  $X_{t_2-1}$ , as  $X_{t_2-1}$  is available from the sample of

the previous batch. The message passing and sampling equations for  $X$  are given in Algorithm 9.

Next, we describe the approach to initializing  $Z$  given a sample of  $X$ . The message passing and sampling procedures for  $Z$  are very similar to those in the initialization procedure of SSIM inference. However, apart from the  $K$  sampled structure and parameter states, we must now also account for the possibility of the arrival of a new state from the prior. Unlike in the first initialization approach, we have already sampled  $X$ , allowing for exact message passing and sampling of  $Z$  for the remainder of the initialization. Computing the message entries for the first  $K$  states is identical to the analogous message passing computation in SSIM inference. Computing the  $K + 1^{\text{st}}$  message entry requires computing  $P(X_{i+1}|X_i; \beta, \gamma)$  instead of  $P(X_{i+1}|X_i, E_{Z_i}, \Theta_{Z_i})$ . This expression was expanded in Equations 3.7 and 3.8, and is restated here for readability:

$$P(X_{t+1}|X_t; \beta, \gamma) = \prod_{i=1}^N \sum_{\tilde{p}a(i)} P(X_{t+1}^i | X_t^{\tilde{p}a(i)}; \gamma) P(\tilde{p}a(i); \beta). \quad (3.30)$$

This computation is also used in the forwards sampling procedure to compute the probability that  $Z_i$  takes on the value of  $K + 1$  at time  $i$ . The full algorithm for approach 2 to initialization is given below.

```

Data:  $t_2, t_3, Y, X_{t_2-1}, Z_{t_2-1}, \{\tilde{E}, \tilde{\theta}\}_{1:K}, \Sigma, R, \pi$ 
Result:  $X_{t_2:t_3}, Z_{t_2:t_3}$ 
// Message passing and sampling  $X$ , assuming  $(A_i, \Sigma_i) = (I, \Sigma) \forall i$ .
 $m^{t_3}(X_{t_3}) = 1;$ 
for  $i = t_2 : t_3 - 1$  do
|  $m^i(X_i) = \int_{X_{i+1}} \mathcal{N}(X_{i+1}; X_i, \Sigma) \mathcal{N}(X_{i+1}; Y_{i+1}, R) dX_{i+1};$ 
end
for  $i = t_2 : t_3$  do
|  $X_i \sim P(X_i | X_{t_2-1:i-1}, Y) \propto \mathcal{N}(X_i; X_{i-1}, \Sigma) \mathcal{N}(X_i; Y_i, R) m^i(X_i);$ 
end
// Message passing and sampling  $Z$ , allowing for  $Z = K + 1$ .
 $m^{t_3}(Z_{t_3}) = 1;$ 
for  $i = t_2 : t_3 - 1$  do
| for  $Z_i = 1 : K$  do
| |  $m^i(Z_i) = \sum_{Z_{i+1}} P(Z_{i+1} | Z_i, \pi) P(X_{i+1} | X_i, E_{Z_i}, \Theta_{Z_i}) m^{i+1}(Z_{i+1});$ 
| end
|  $m^i(K + 1) = \sum_{Z_{i+1}} P(Z_{i+1} | K + 1, \pi) P(X_{i+1} | X_i; \beta, \gamma) m^{i+1}(Z_{i+1});$ 
end
for  $i = t_2 : t_3$  do
|  $Z_i \sim P(Z_i | Z_{t_2-1:i-1}, X, \tilde{E}, \tilde{\theta}, \pi) \propto P(Z_i | Z_{i-1}, \pi) P(X_i | X_{i-1}, Z_i) m^i(Z_i);$ 
end

```

**Algorithm 9.** Batch initialization of  $X$  and  $Z$ : Approach 2. We use the  $t_2, t_3$  notation for the sake of consistency with Algorithm 3.3.

### ■ 3.7 Gibbs Sampler

After initialization, we must run rounds of Gibbs sampling to be able to take samples from the true distribution. The sampling procedure of  $\pi, \omega$ , and  $\{\tilde{E}, \tilde{\theta}\}_{1:K}$  is identical to that in SSIM. The sampling steps for  $X$  and  $Z$  are almost identical to those in SSIM, except for the fact that every time point we wish to sample has a time point immediately before it as well. This allows us to always sample from conditional distributions given



previous values instead of from an initial distribution. Specifically, we now sample:

$$X_{t_1} \sim P(X_{t_1}|X_{t_1-1}, E_{Z_{t_1}}, \Theta_{Z_{t_1}})P(Y_{t_1}|X_{t_1})m^{t_1}(X_{t_1}) \text{ and} \quad (3.31)$$

$$Z_{t_1} \sim P(Z_{t_1}|Z_{t_1-1}, \pi)P(X_{t_1}|X_{t_1-1}, E_{Z_{t_1}}, \Theta_{Z_{t_1}})m^{t_1}(Z_{t_1}) \quad (3.32)$$

instead of

$$X_{t_1} \sim P(X_{t_1})P(Y_{t_1}|X_{t_1})m^{t_1}(X_{t_1}) \text{ and} \quad (3.33)$$

$$Z_{t_1} \sim P(Z_{t_1})P(X_{t_1}|X_{t_1-1}, E_{Z_{t_1}}, \Theta_{Z_{t_1}})m^{t_1}(Z_{t_1}). \quad (3.34)$$

A full characterization of the post-initialization Gibbs sampling procedure is given below in Algorithm 10, which we refer to throughout this thesis as  $A_2$ .

**Data:**  $t_2, t_3, Y_{t_2:t_3}, X_{t_2-1:t_3}, Z_{t_2-1:t_3}, \{\tilde{E}, \tilde{\theta}\}_{1:K}$   
**Result:**  $X_{t_2:t_3}, Z_{t_2:t_3}, \{\tilde{E}, \tilde{\theta}\}_{1:K}$   
**for**  $i = 1 : N$  **do**  
    Sample  $\pi \sim P(\pi|Z_{t_1-1:t_3}; \alpha)$ ;  
    Sample  $\{\tilde{E}, \tilde{\theta}\}_{1:K} \sim P(\{\tilde{E}, \tilde{\theta}\}_{1:K}|Z_{t_1:t_3}, X_{t_1-1:t_3}; \beta, \gamma)$ ;  
    Sample  $\omega \sim P(\omega|X_{t_1:t_3}, Y_{t_1:t_3}; \delta)$ ;  
    Sample  $X_{t_1:t_3} \sim P(X_{t_1:t_3}|X_{t_1-1}, Z_{t_1:t_3}, Y_{t_1:t_3}, \{\tilde{E}, \tilde{\theta}\}_{1:K})$ ;  
    Sample  $Z_{t_1:t_3} \sim P(Z_{t_1:t_3}|Z_{t_1-1}, X_{t_1:t_3}, \{\tilde{E}, \tilde{\theta}\}_{1:K}, \pi)$ ;  
**end**

**Algorithm 10.**  $A_2$ : post-initialization Gibbs sampler.

### ■ 3.8 Summary

In this chapter, we introduced and developed ONSTIM, a procedure for performing inference in an online nonparametric switching temporal interaction model. We proposed an online approach to inference that incorporates data incrementally and performs inference only on the latest batch of time points. A nonparametric approach to selecting the number of transition models was developed that operates by allowing the proposal of new states during the initialization of the Gibbs samplers, and that instantiates a new model if a new state is indeed proposed.

Drawbacks of both developments were also discussed. Subdivision of the time series into batches prevents the use of information from time points that are further away

than the specified batch length, introducing a new limitation compared to the original algorithm of Dzunic and Fisher. Our approach to sampling new states relies heavily on sampling instead of analytical approaches, thereby introducing significant variance, and also can only operate during the initialization phase of the Gibbs samplers. Analytical approximation approaches and their drawbacks were also discussed, and they were ultimately not implemented, though that remains a potential avenue for further work.

Next, two different initialization approaches for the Gibbs samplers were described, along with the relative merits and weaknesses of each one. We now point out that for experimental purposes, we will exclusively use initialization approach 2. Finally, we described a slightly modified version of the original Gibbs sampling procedure for sampling in ONSTIM. Experimental results of ONSTIM on both synthetic and real financial datasets will be reported in Chapter 4.

# Results

In this chapter, we will provide experimental results from the application of ONSTIM to synthetic and real datasets. First, we will empirically characterize the performance of ONSTIM through its application to synthetic datasets with known true values for all variables in the graphical model. Knowing the true values of variables in the synthetic datasets provides us a ground truth against which to compare the results of ONSTIM, thereby providing a metric for its performance. We will describe the process of generating these synthetic datasets, and then explore the behavior of ONSTIM on these datasets as we vary procedural parameters, such as the batch offset and the values of  $\alpha_{self}$  and  $\alpha_{new}$ .

Then, we will apply ONSTIM to financial datasets, constructed from time series of the prices of various United States stocks and other financial instruments. Specifically, we will apply ONSTIM to both interday and intraday datasets.

### ■ 4.1 Empirical Model Characterization

In this section, we present empirical results of ONSTIM on synthetic datasets with the aim of characterizing the behavior of ONSTIM in a variety of parameter regimes. We are interested in the performance of ONSTIM as we vary the batch offset,  $\alpha_{new}$ , and  $\alpha_{self}$ . We wish to characterize the performance of ONSTIM with respect to the *parameters* of the generative model described in the previous chapter instead of with respect to any particular dataset that is an instantiation of the generative model. Thus, we will effectively marginalize over the specific dataset instantiation by sampling several realizations for each generative model parameter set, running ONSTIM on each realization, and averaging results over each of the realizations to obtain an estimate of the

distribution of the performance of ONSTIM for given parameter conditions.

We will first describe the methods by which the synthetic datasets were generated. We will then display and provide exposition of summary statistics of the performance of ONSTIM on the synthetic datasets. We are specifically interested in the ability of ONSTIM to accurately detect the number of switching states in the generated datasets, and in how this ability varies with the choice of parameters. Finally, we will discuss the implications of the results and attempt to shed some light on the reasons that ONSTIM behaves as it does.

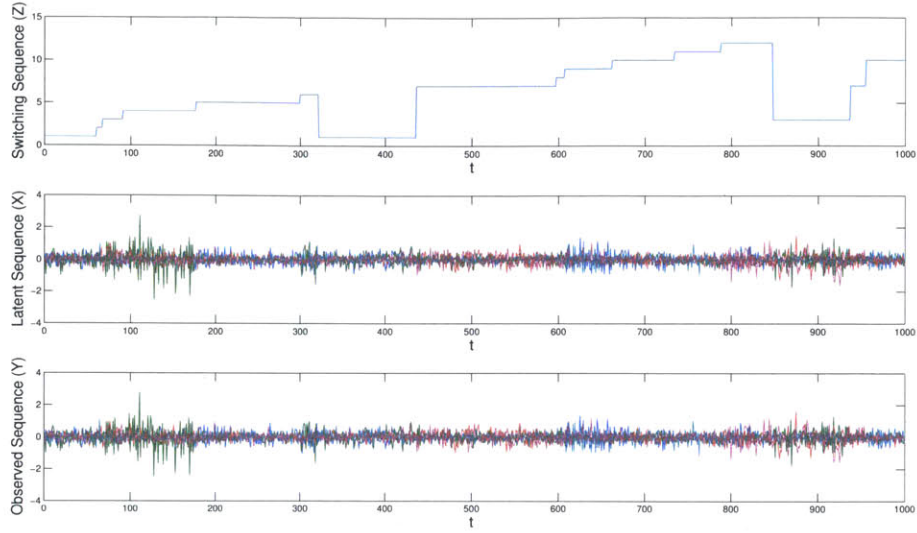
### ■ 4.1.1 Synthetic Dataset Generation

Synthetic datasets were generated by sampling a switching sequence from a transition model specified by  $\alpha_{self}$  and  $\alpha_{new}$  as described above. All switching sequences were initialized to  $K = 1$ , so there was no need for an initial distribution of  $Z$ . For each sampled instance of  $Z$ ,  $K$  interaction models and corresponding parameter sets were sampled from the parent set prior on structures and the matrix-normal inverse-Wishart prior on parameters. The value of the latent sequence  $X$  at time 0 was sampled uniformly at random from the interval  $[0, .001]$ , and subsequent samples were taken according to the structure and parameter model indexed by the value of  $Z$  at that time. Finally, the observation noise covariance parameter  $\xi$  was sampled from inverse-Wishart prior. The observation sequence  $Y$  was then sampled conditioned on the sampled values of  $X$  and  $\xi$ . The number of nodes in all datasets was set to 5, and the maximum allowed parents were 3.

Synthesized datasets were filtered to prevent occurrences of destabilizing exponential growth in the latent or observed sequences. This was achieved heuristically by disallowing datasets in which the observed sequence took values at any time point with absolute value greater than 100. A sample synthetic dataset generated is shown in Figure 4.1.

### ■ 4.1.2 Inferred Number of Switching States

An important metric of the performance of ONSTIM is the relationship between the inferred number of states and the true number of states, whose values we know in the case of synthetic datasets. We will refer to the empirical distribution of the total number of states from the sampling procedure as  $\hat{K}$ , which is an estimator of the true



**Figure 4.1.** Sample synthetic dataset with  $\alpha_{new} = 20$ ,  $\alpha_{self} = 2500$ . Switching, latent, and observed sequences are shown.

underlying value of  $K$ .

First, we computed empirical conditional distributions of  $\hat{K}$  given  $K$ . These distributions were computed for  $(\alpha_{new}, \alpha_{self}) = (5, 500), (10, 500), (20, 2500),$  and  $(40, 2500)$ , and for batch offset values of 2 and 10. For each transition model parameter, 20 datasets of length  $T = 1000$  were generated, and 25 samples were computed for each dataset, i.e.  $S = 25$ . Within each batch sample, 9 samples were skipped to allow for burn-in and the tenth sample was taken from the sample chain. To compute probabilities conditional on the prior over structures and parameters, averaging was done over 40 structure/parameter models which were sampled from the prior on interaction structures and transition parameters. The transition model parameters in the inference algorithm were chosen to match the parameters in the corresponding generative model. The conditional distributions of  $\hat{K}$  given  $K$  for all eight parameter conditions are shown below in Figures 4.2 and 4.3.

Next, we computed the conditional bias of the estimator  $\hat{K}$  given that  $K$  takes on



a fixed value  $k$ , which we can write as  $\text{Bias}_k(\hat{K})$ :

$$\text{Bias}_k(\hat{K}) = \mathbb{E}_{\hat{K}|k}[k - \hat{K}] = \mathbb{E}_{\hat{K}|k}[\hat{K}] - k \quad (4.1)$$

$$= \sum_{\hat{k}} [\hat{k}P(\hat{K} = \hat{k}|K = k)] - k. \quad (4.2)$$

The conditional bias of  $\hat{K}$  given  $K$  is a function of  $K$ . Next, we will use the conditional bias to compute the global bias of the estimator  $\hat{K}$ , also commonly just called the bias, which marginalizes out the distribution of  $K$ .

$$\text{Bias}_K(\hat{K}) = \mathbb{E}_K[\text{Bias}_k(\hat{K})] \quad (4.3)$$

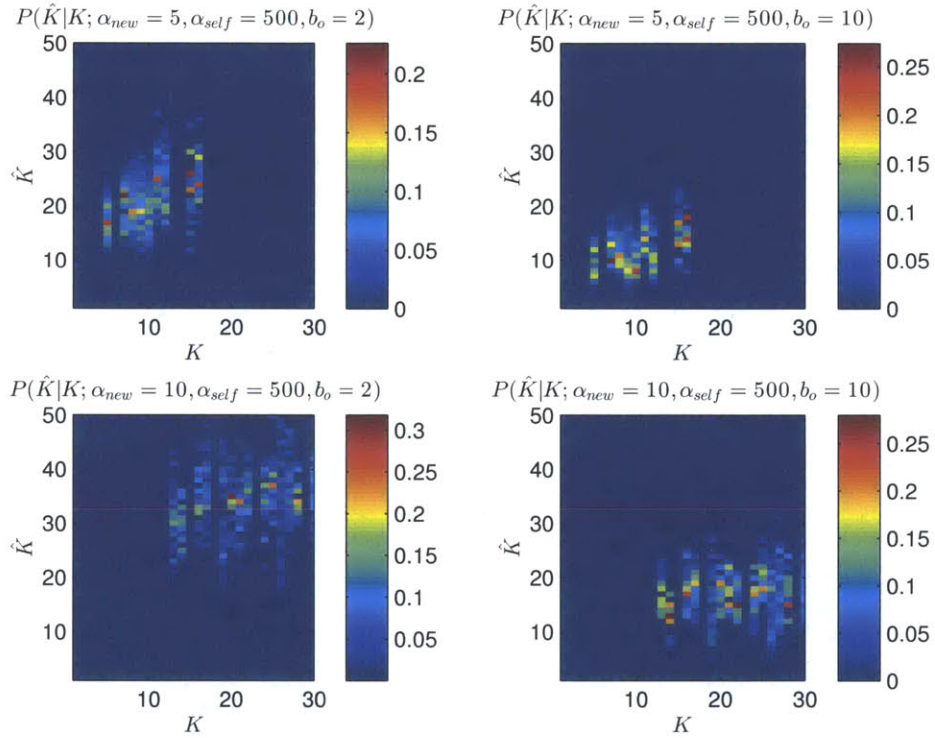
$$= \sum_k P(K = k) \left[ \sum_{\hat{k}} \hat{k}P(\hat{K} = \hat{k}|K = k) - k \right]. \quad (4.4)$$

Other possible characterizations of the accuracy of inferring the true number of states could include looking at the average difference between the true value of  $K$  and the median, mode, or other central tendency statistic of the distribution of  $\hat{K}$  given  $K$ . However, we will limit ourselves to considering bias for this purpose. Conditional and global biases are shown in Figures 4.4 and 4.5, with global biases listed in the title of the figures for the respective conditional bias.

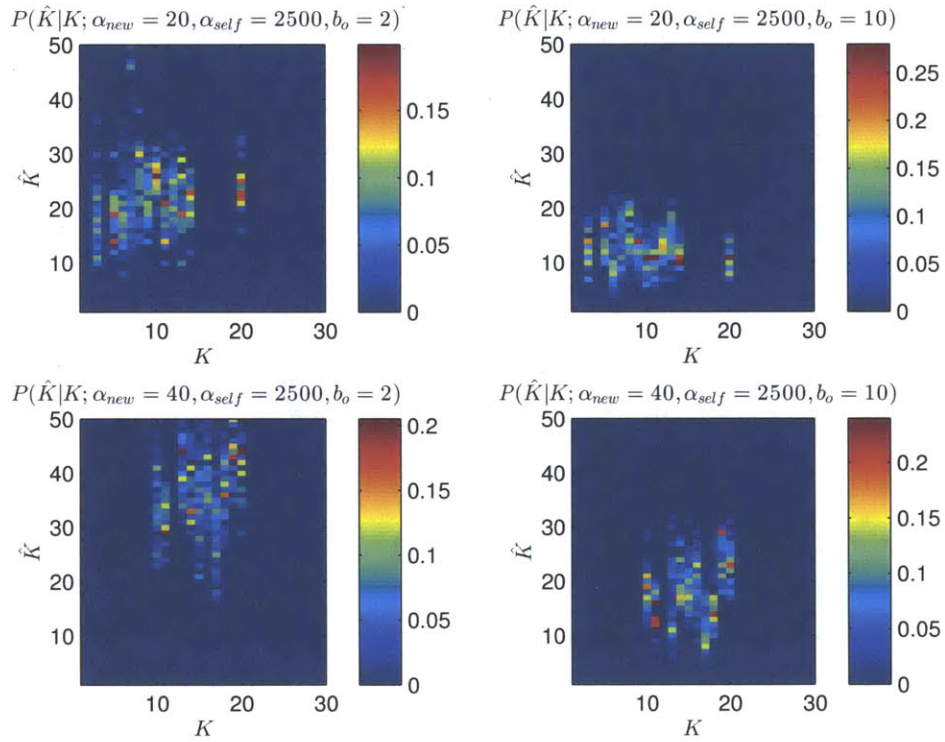
### ■ 4.1.3 Discussion

The choice of batch offset has a significant effect on the number of inferred states. Specifically, experiments run with a batch offset of 2, the lower of the two values used in experiments, display instantiation of significantly more states than those run with a batch offset of 10. Furthermore, experiments run with a batch offset of 2 display a positive conditional bias for every setting of the parameters  $\alpha_{new}$  and  $\alpha_{self}$ , and for every true value of  $K$ . Not only is the conditional bias consistently positive for a batch offset of 2, but Figures 4.2 and 4.3 suggest that in almost no samples is  $\hat{K}$  less than or equal to the true value of  $K$ . To summarize, experiments run with a batch offset of 2 consistently and significantly overestimate the number of states in the dataset.

A possible explanation for the much higher number of states instantiated in experiments with batch offset 2 than in those with batch offset 10 may simply be the fact that since approximately 5 times as many batches are sampled with offset 2 than with offset 10, ONSTIM has 5 times as many opportunities in which to propose new states. Note

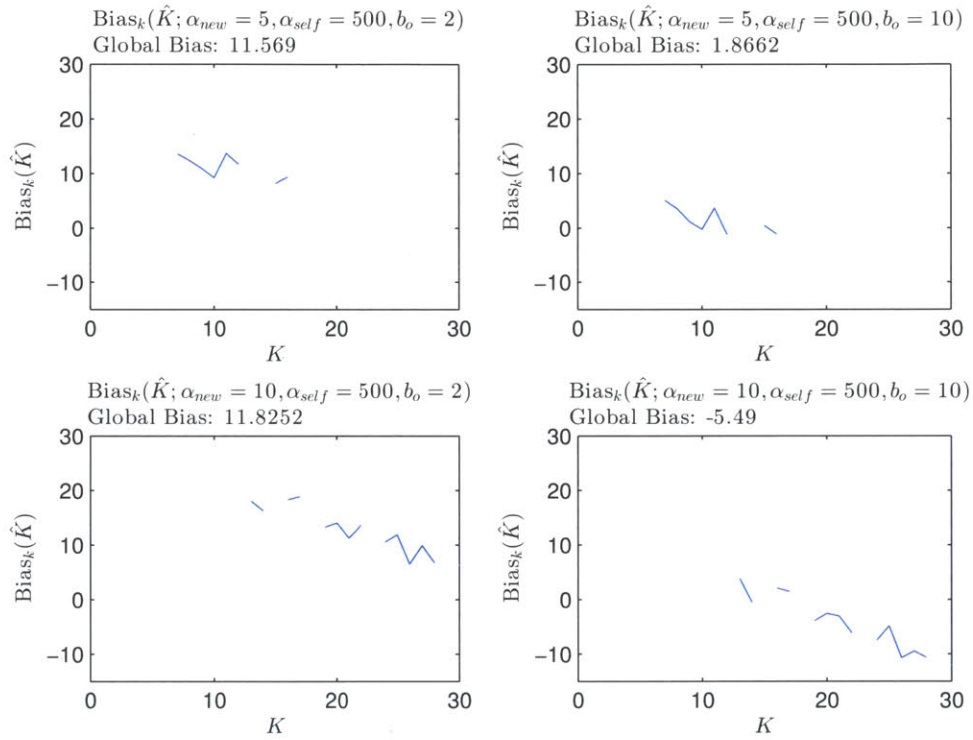


**Figure 4.2.**  $P(\hat{K}|K)$  for low  $\alpha_{new}, \alpha_{self}$ : Distribution of the estimator  $\hat{K}$  given  $K$ , for parameter values  $(\alpha_{new}, \alpha_{self}) = (5, 500), (10, 500)$ . The distribution of  $\hat{K}$  for a given value of  $K$  should be interpreted by taking the column above  $K$  as the distribution of interest. If a certain value of  $K$  was not sampled in any dataset, the entire conditional distribution column for that  $K$  is displayed as 0. Note that the estimator  $\hat{K}$  is significantly higher when the batch offset is chosen to be 2 than when it is chosen to be 10.

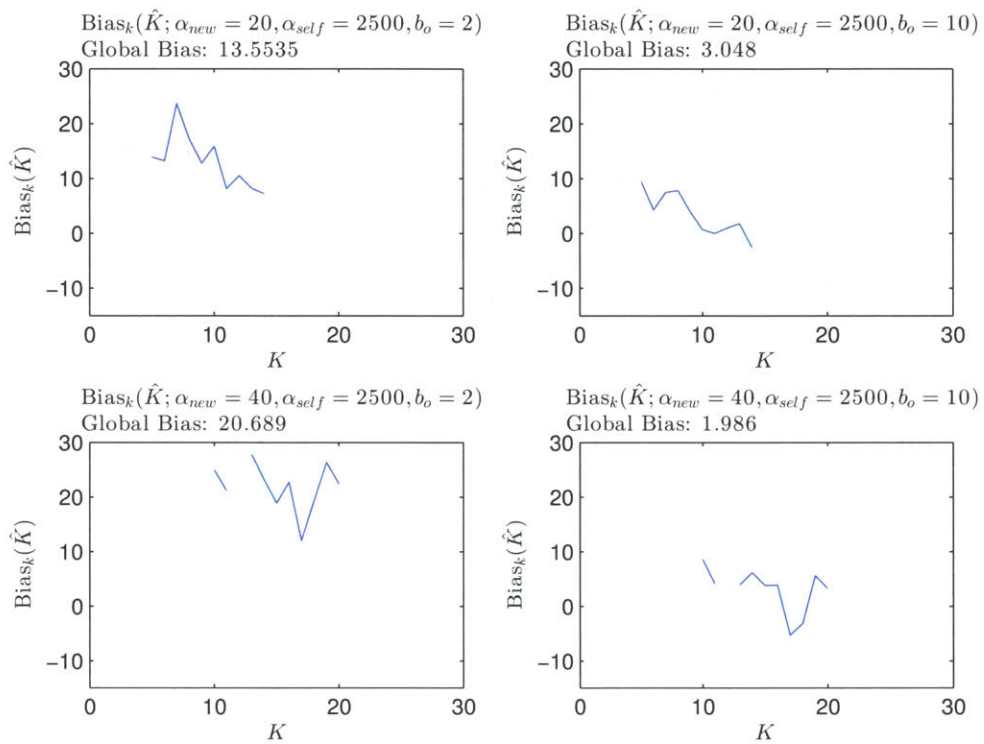


**Figure 4.3.**  $P(\hat{K}|K)$  for high  $\alpha_{new}, \alpha_{self}$ : Distribution of the estimator  $\hat{K}$  given  $K$ , for parameter values  $(\alpha_{new}, \alpha_{self}) = (20, 2500), (40, 2500)$ . The figure can be interpreted in the same manner as Figure 4.2. Again, the estimator  $\hat{K}$  is significantly higher when the batch offset is chosen to be 2 than when it is chosen to be 10.





**Figure 4.4.** Conditional biases of  $\hat{K}$  given  $K$  and global biases of  $\hat{K}$ , for parameter values  $(\alpha_{new}, \alpha_{self}) = (5, 500), (10, 500)$ . A positive bias implies that  $\hat{K}$  overestimated  $K$ , while a negative bias implies that  $\hat{K}$  underestimated  $K$ . Note that the sample average  $\bar{K}$  can be recovered from the graph through the relationship  $\bar{K} = K + \text{Bias}_K(\hat{K})$ . Bias as a function of  $K$  tends to be downwards sloping.



**Figure 4.5.** Conditional biases of  $\hat{K}$  given  $K$  and global biases of  $\hat{K}$ , for parameter values  $(\alpha_{new}, \alpha_{self}) = (20, 2500), (40, 2500)$ . The figure can be interpreted in the same manner as Figure 4.4.

that in all experiments,  $\alpha_{new}$  was chosen to be greater than 1, implying that ONSTIM is more likely to instantiate a new state to explain an incoming observation than to reuse an existing interaction model that explains the observation as well as the base measure does. If such situations occur frequently, running ONSTIM with a batch offset of 2 does allow for significantly more opportunities for the instantiation of new states.

Another potential scenario that could help explain the higher number of states instantiated by batch offset 2 than by batch offset 10, but not the consistently high bias, is the possibility that two or more new interaction structures arrive in rapid succession. In such a scenario, setting the batch offset to 10 would result in different new states getting instantiated as the same new model, while keeping the batch offset at a lower level like 2 would allow for greater granularity in detecting separate new states and instantiating them individually.

Unrelated to differences in batch offset, the generally downwards slope in most of the conditional bias figures (Figures 4.4 and 4.5) also prompts some interesting questions. The higher the true number of states  $K$ , the less the bias of the estimator  $\hat{K}$  tends to be. While it is certainly evident from Figures 4.2 and 4.3 that higher values of  $K$  tend to correspond to higher values of  $\hat{K}$ , the negative slope of the bias implies a tendency of  $\hat{K}$  to move towards a central value determined by the parameters of the inference algorithm. It would be interesting to try expressing the distribution of  $\hat{K}$  in terms of the true value of  $K$  for an instantiated dataset and in terms of the parameters  $\alpha_{new}$  and  $\alpha_{self}$ . However, exploring this phenomenon further would require obtaining many more  $(K, \hat{K})$  data points.

## ■ 4.2 Experiments with Financial Datasets

Financial datasets comprise a rich and complex field for experimentation with graphical model inference procedures. Sets of financial instruments often have a wide variety of conditional dependence relationships structures. In contrast to the synthetic datasets whose experiments allowed for validation and testing of the performance ONSTIM, financial datasets do not have a corresponding generative model from which we can know the “true structure”, at least not a model that is directly observable. Consequently, it is difficult to directly evaluate the performance of ONSTIM on financial datasets, although the predictive capabilities of the inference results of ONSTIM could serve as

a proxy for result accuracy.

However, since ONSTIM employs a sampling approach, after performing inference we are equipped with a characterization of the full joint distribution, which allows us to ask a wide variety of interesting questions about the dataset, such as the probability of the existence of an edge during a given time window, or the probability that the switching states sequence took on the same value at two different times. This allows us to shed light onto the interaction structures present in the datasets, and especially lends us insight into the process of arrival of new states in the datasets of interest. For example, we may be interested in knowing which financial instruments are the most influential in a given period of time, or how likely it is that a new interaction structure arose during that time. We will now address these and other questions for a interday S&P100 dataset, a particular intraday S&P100 dataset, and an interday S&P500 sector dataset.

For the sake of consistency, most parameters of ONSTIM were set to the same values as in the synthetic dataset experiments. Because of increased dimensionality of the data vectors, however, the number of maximum parents was reduced to 2 for all datasets. Furthermore, all financial dataset experiments were conducted with a batch offset of 10, as this was shown in the synthetic experiments to yield less biased estimators of state number in our parameter regimes of operation. All other parameters were kept the same.

#### ■ 4.2.1 S&P100: 2007-2012

The S&P100 is a stock index consisting of the 100 largest publicly traded companies in the United States. We have constructed two datasets with the stocks from the S&P100. The first dataset, which we consider here, is a long-term interday dataset that was constructed using the prices at the close of each trading day from September 4, 2007, to December 31, 2012. Since the S&P100 was rebalanced over our time window of interest, we restrict ourselves to consideration of stocks that were members of the S&P100 from September 4, 2007, through December 31, 2012, yielding a list of 90 stocks. As is typical when working with financial datasets, instead of directly using prices to track the movements of a stock, we use log returns of the prices as our observed time series in this dataset and all subsequent ones.

Results from inference on this dataset exhibit significant new instantiation, as dis-

played in Figure 4.6. Figure 4.8 displays the switching similarity matrices, which displays in entry  $(i, j)$  the probability taken across all samples that the interaction structures and parameters active at times  $i$  and  $j$  are the same, i.e. that  $Z_i = Z_j$ . Since similar switching states may be indexed differently across different samples, we can only reason about the equality of switching states that coexist within a single sample. Figure 4.8 shows that the market was largely in the same state for the majority of the duration of the experiment. However, for a significant chunk of time from the latter part of 2008 through the beginning of 2009, the market appears to have been in a different set of states that varied somewhat rapidly over this course of time. It is possible that this deviation from the normal market state corresponds to impact felt in the equity markets from the financial crisis of 2008. A second visible, albeit much smaller, deviation occurs in several samples for all parameter values around  $t = 1000$ , corresponding to the end of 2011 at the beginning of 2012. It is not obvious what market event, if any, this transition corresponds to.

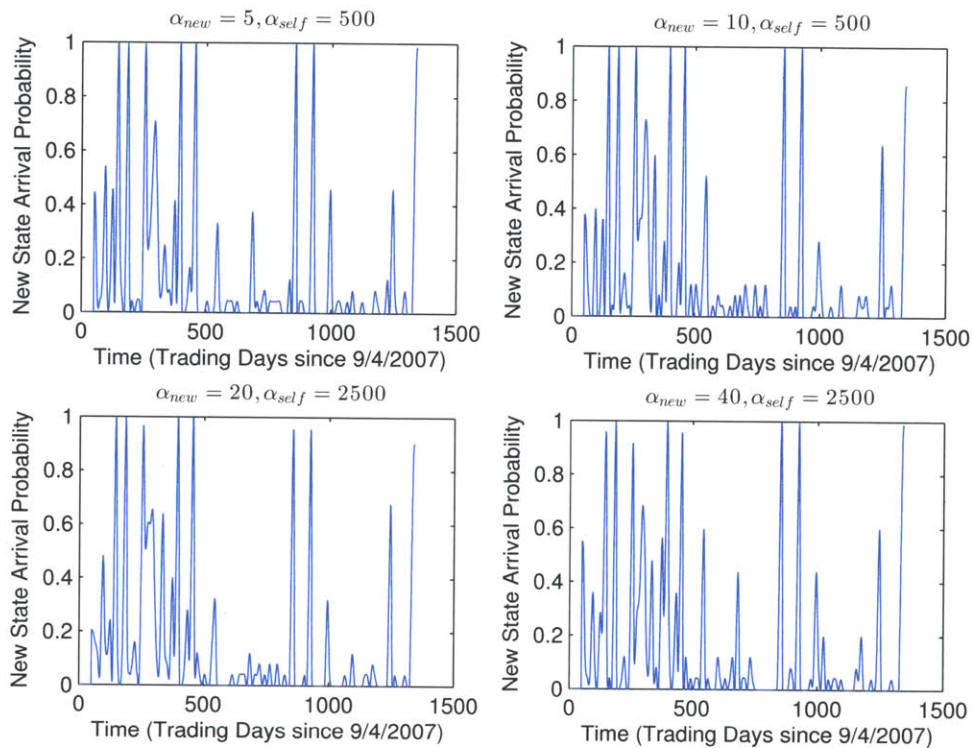
#### ■ 4.2.2 S&P100: Flash Crash

The second S&P100 dataset that we will look at is an intraday dataset from the day of May 6, 2010, that was constructed with the last trade prices taken every minute. May 6, 2010, was the day of the Flash Crash, during which the US stocks rapidly dropped during the afternoon. The S&P500 lost 5% of its value in a few minutes, and subsequently quickly recovered back to its original price [6]. The Flash Crash represents a notable yet short-lived market phenomenon, and thus presents an interesting scenario for ONSTIM.

Figure 4.9 shows the probability of new state arrivals. All samples in all parameter settings sampled the Flash Crash as the arrival of a new state. Figures 4.11 and 4.12 show that the market was largely in a single state until the Flash Crash, at which point the system appears to have rapidly transitioned between several different states, none of which were the same as the original state. Towards the close of the day, the observations suggest a return to a somewhat recurrent state, which most samples assign to a different interaction model index than the original state at the beginning of the day.

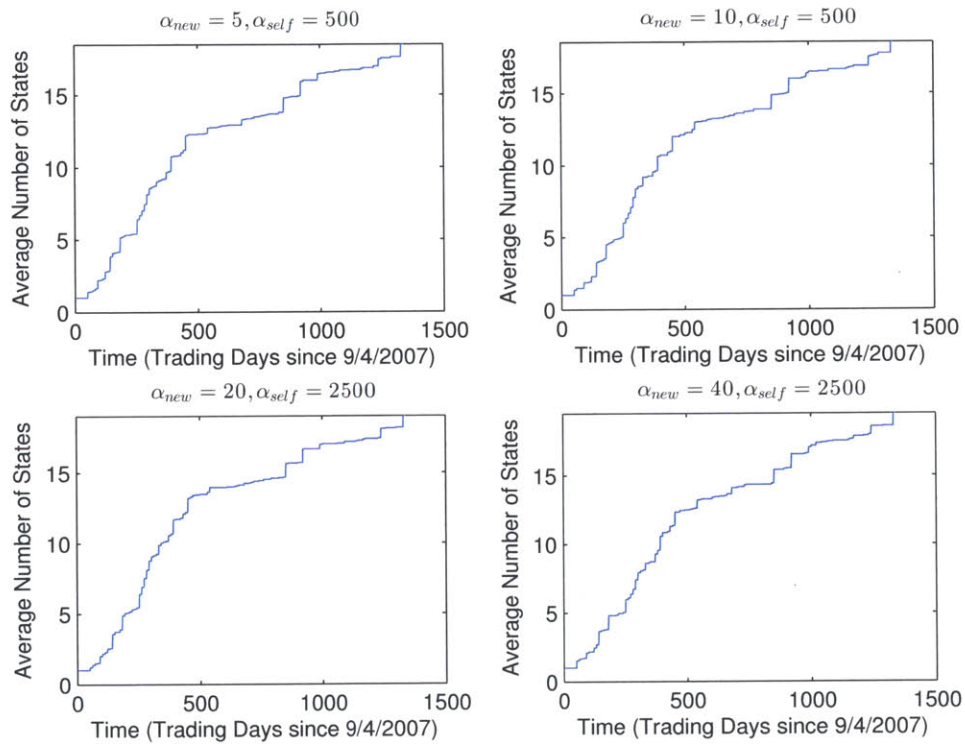
Figures 4.11 and 4.12 also demonstrate the ability of ONSTIM to perform both filtering, corresponding to a batch lag of 0, and fixed-lag smoothing corresponding to



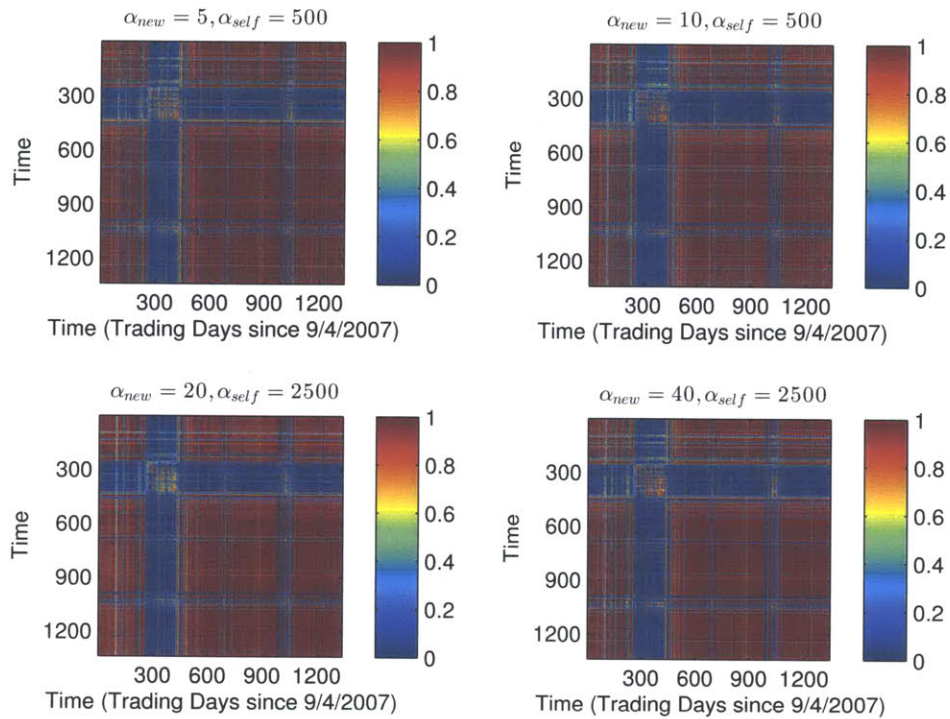


**Figure 4.6.** S&P100 Interday: New state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . Note that the value of the graph at time  $t$  is not necessarily the probability of a new state instantiation at time  $t$ , but rather the probability of the event that the *batch* in which  $t$  appeared instantiated a new state. Almost all samples in all parameter settings agree on the instantiation of a new state at 8 points, visible as peaks with probability close to 1.

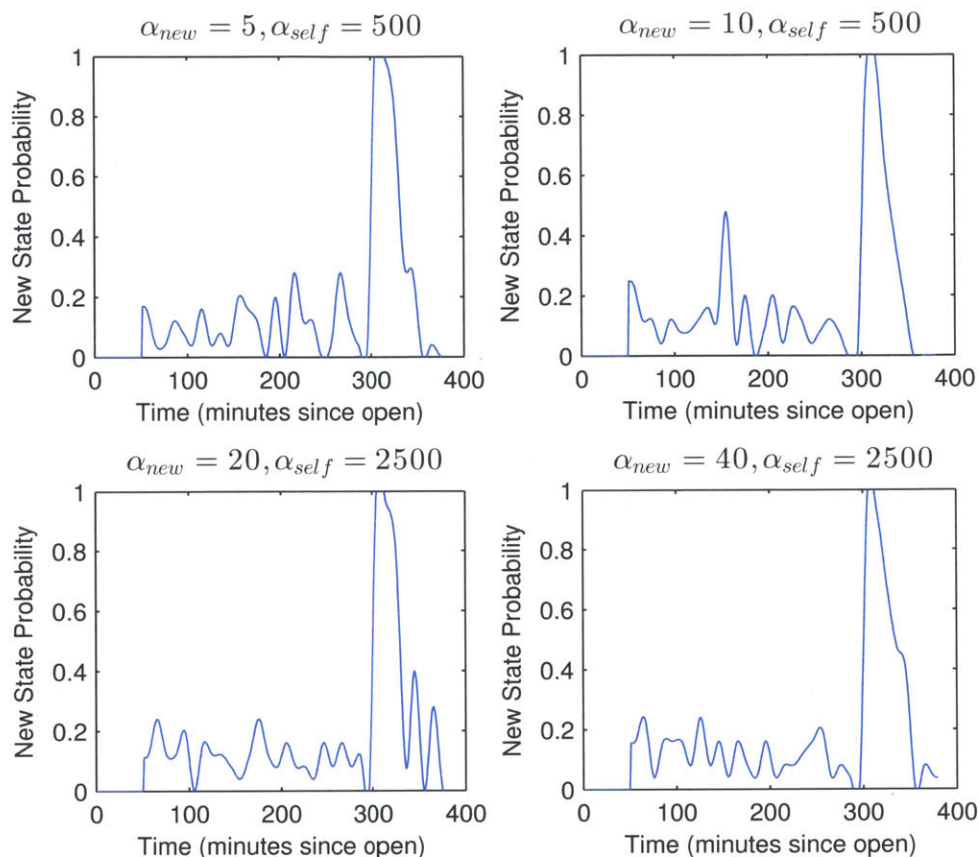




**Figure 4.7.** S&P 100 Interday: The total instantiated numbers of states state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . This figure can be interpreted as the integral of Figure 4.6.



**Figure 4.8.** S&P100 Interday: Switching similarity matrix. Entry  $(i, j)$  indicates the probability taken across all samples that  $Z_i = Z_j$  within a random sample. From the latter part of 2008 through the beginning of 2009, the figure suggests that the market went through set of states that varied somewhat rapidly. This switching sequence may correspond to the effect of the 2008 financial crisis on US equity markets.

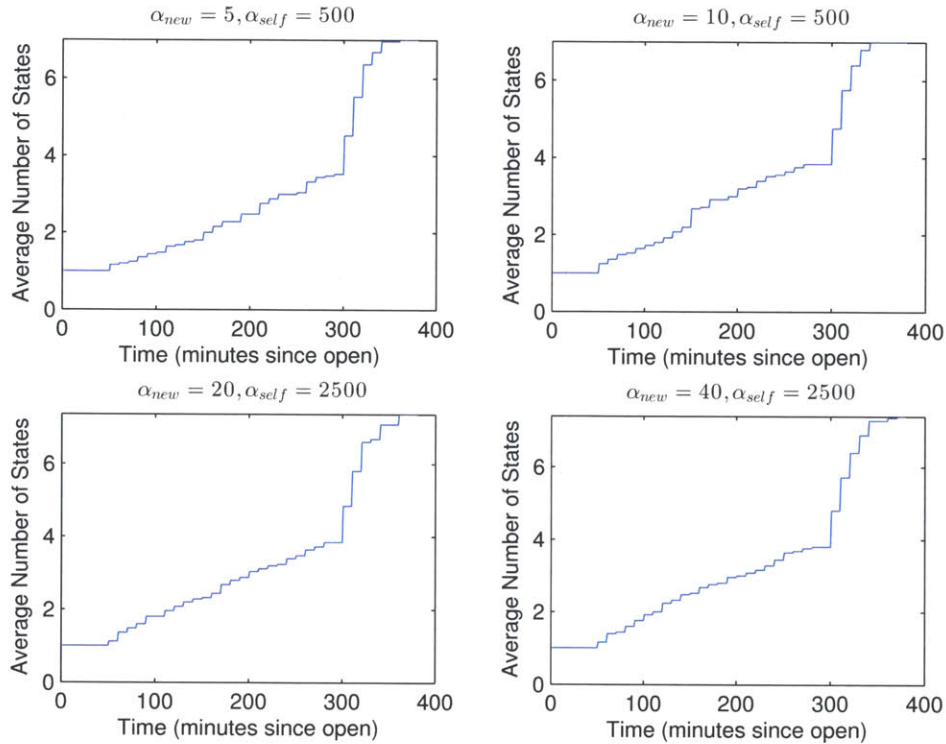


**Figure 4.9.** Flash Crash: New state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . All parameter values recognize the arrival of a new state a little more than 5 hours into the trading day, corresponding to the known start of the crash.

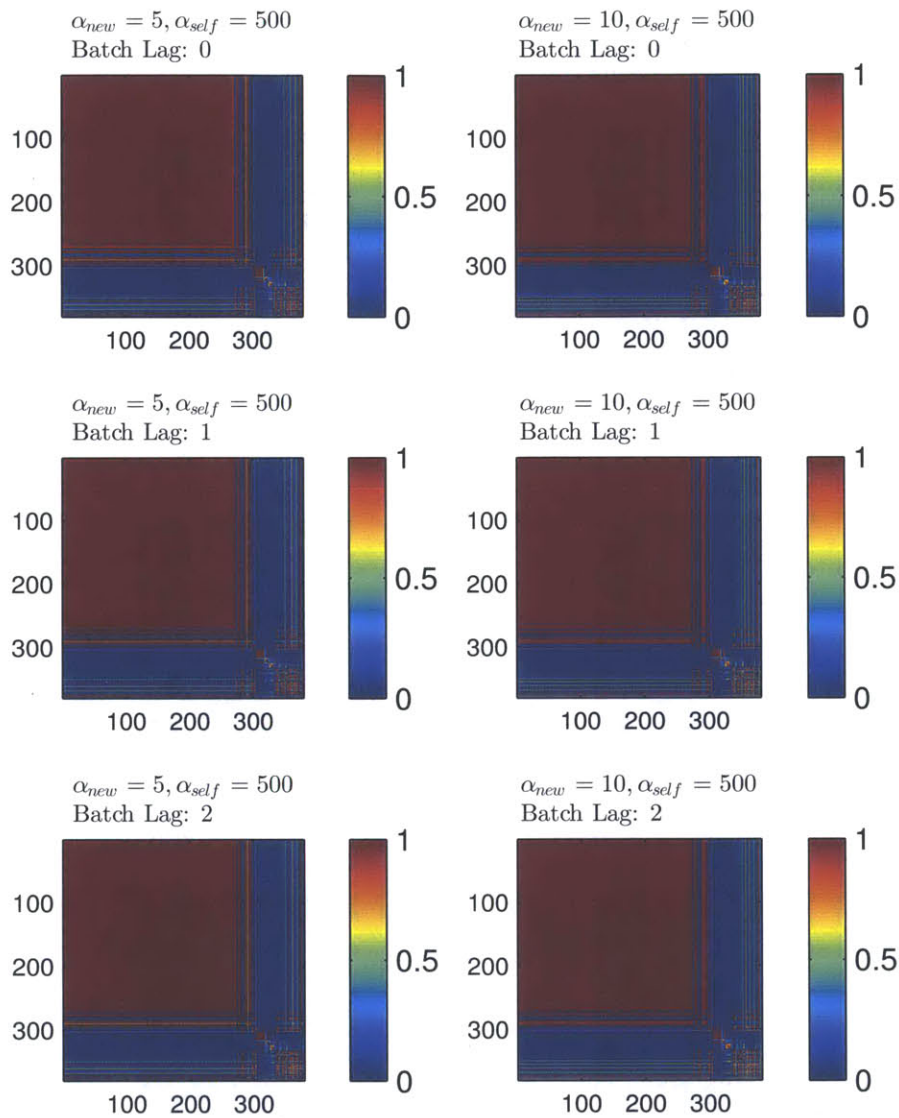
positive batch lags. In the context of ONSTIM, filtering entails evaluating a quantity of interest using the sample from the batch in which time  $t$  first appears as the sample at time  $t$ . Fixed-lag smoothing with a batch lag of  $\lambda$  refers to using the sample from the  $\lambda + 1^{\text{st}}$  batch in which  $t$  appears. In the case of the switching similarity matrices displayed, however, there does not appear to be significant difference between the results from filtering and fixed-lag smoothing for lags of 1 and 2.

### ■ 4.2.3 S&P Sector ETFs

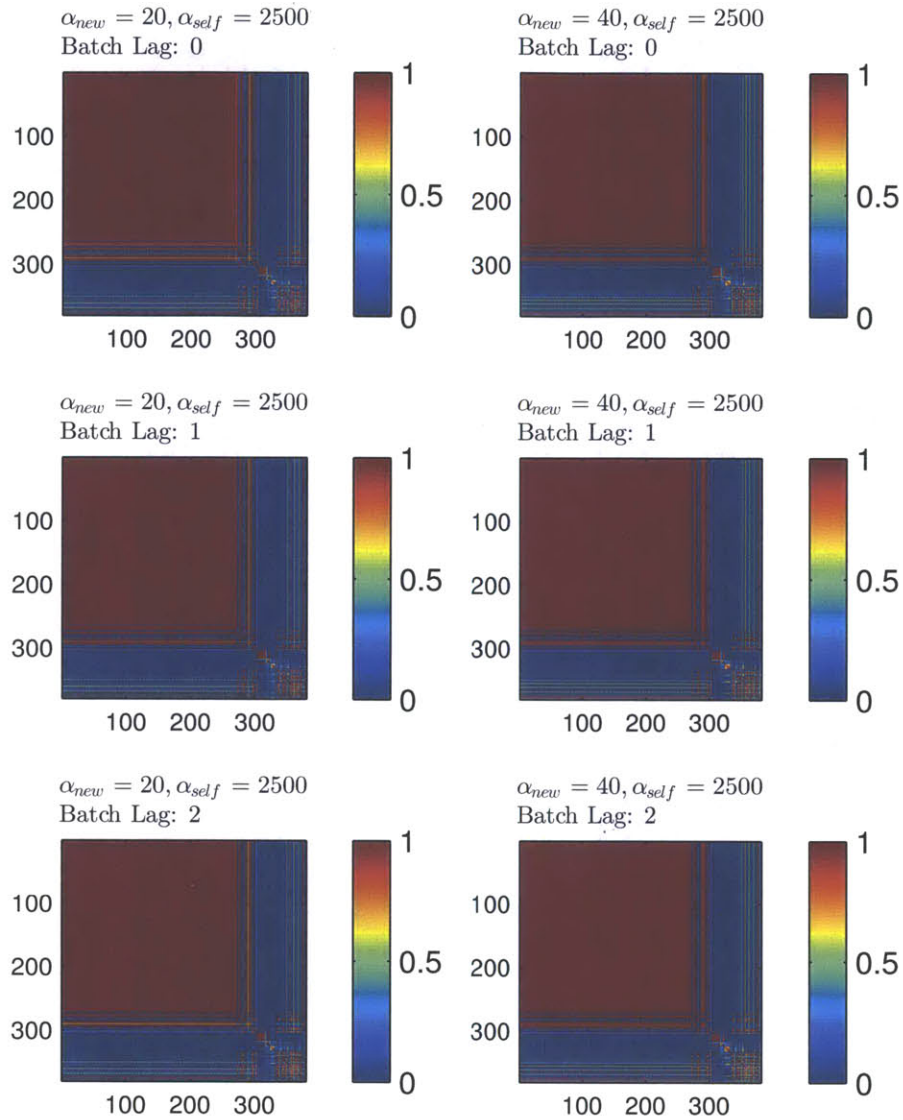
Next, we will consider datasets constructed from S&P sector exchange traded funds (ETFs). The S&P500 index, a superset of the S&P100, is the most common bench-



**Figure 4.10.** Flash Crash: The total instantiated numbers of states state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . Note how the arrival rate of structure/parameter states jumped at around the time of the Flash Crash, implying that market interaction structures were moving much faster than minutes before. Note also that all parameter values eventually instantiate approximately 7 states, serving as a marker of the insensitivity of inference on this particular dataset to the choice of transition model parameters.



**Figure 4.11.** Flash Crash: Switching similarity matrices for low values of  $(\alpha_{new}, \alpha_{self})$  are shown. We show switching similarity matrices for batch lags of 0, 1, and 2, displaying the ability of ONSTIM to perform filtering or fixed-lag smoothing for small fixed lags. All batch lags show a static interaction structure throughout the day until approximately 5 hours in, at which point the market rapidly iterates over a number of structures.



**Figure 4.12.** Flash Crash: Switching similarity matrices for high values of  $(\alpha_{new}, \alpha_{self})$  are shown. Again, the results of this experiment appear relatively insensitive to the choice of  $\alpha_{new}$  and  $\alpha_{self}$ .

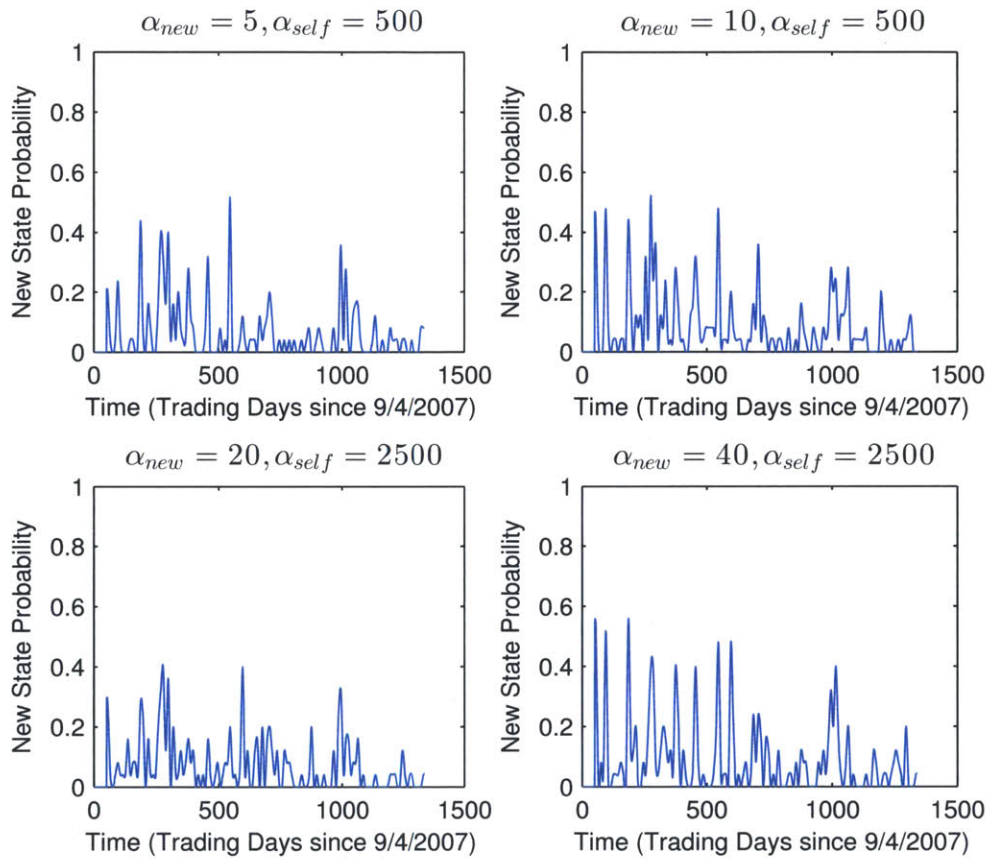


mark index for the United States economy. It is broken down into 9 industry sectors: materials, energy, financials, industrials, technology, consumer staples, utilities, health care, and consumer discretionary. These sectors are referred to with the ticker symbols XLB, XLE, XLF, XLI, XLK, XLP, XLU, XLV, and XLY respectively. The set of 9 sectors serves as a coarse representation of the dynamics of the S&P500, but provides a much more tractable dataset on which to perform inference quickly. For the sake of consistency, we ran ONSTIM on the sector datasets for the same times as the S&P100 datasets.

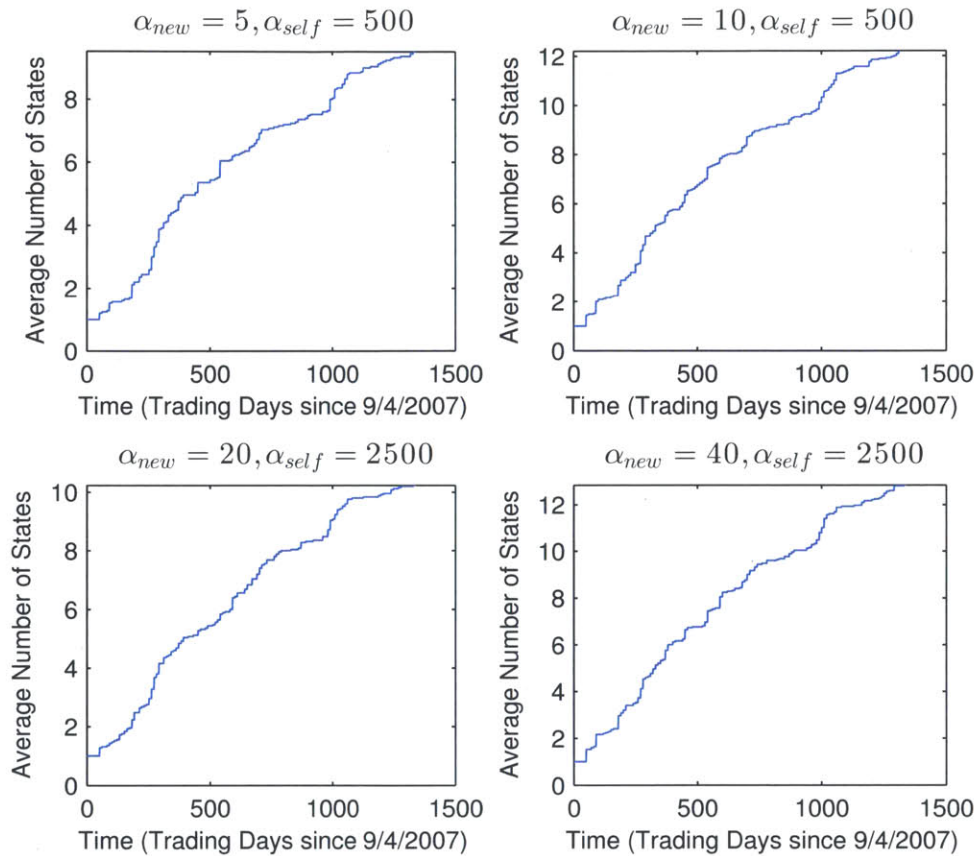
In Figure 4.13, no single day is indicated as having greater than 60% chance of new state instantiation, in contrast to the corresponding Figure 4.6, which shows high instantiation probabilities for the S&P100 interday dataset over the same time period. However, Figure 4.15 tells a state transition story very similar to that of Figure 4.8, the corresponding figure for the S&P100 interday dataset.

Due to the small number of observed variables in the sector ETF datasets, we can display the posterior distribution over edges between the 9 ETFs, shown in Figure 4.16, and derive a meaningful interpretation from the result. The value of the matrix at entry  $(i, j)$  is the probability taken across all samples and times that object  $i$  is a parent of object  $j$ . Self edges are not displayed, as are fixed to occur by default in all interaction structures. For parameter values of  $(\alpha_{new}, \alpha_{self}) = (5, 500)$  and  $(20, 2500)$ , which correspond to lower estimated values of  $K$ , the posterior probability is quite high that XLE, the energy sector ETF, is a parent of XLF, the financial sector ETF. The posterior probability of the existence of the edge  $XLE \rightarrow XLF$  is approximately 3% for both of the mentioned parameter values, which is approximately double the probability of an average edge and the highest of any as well.

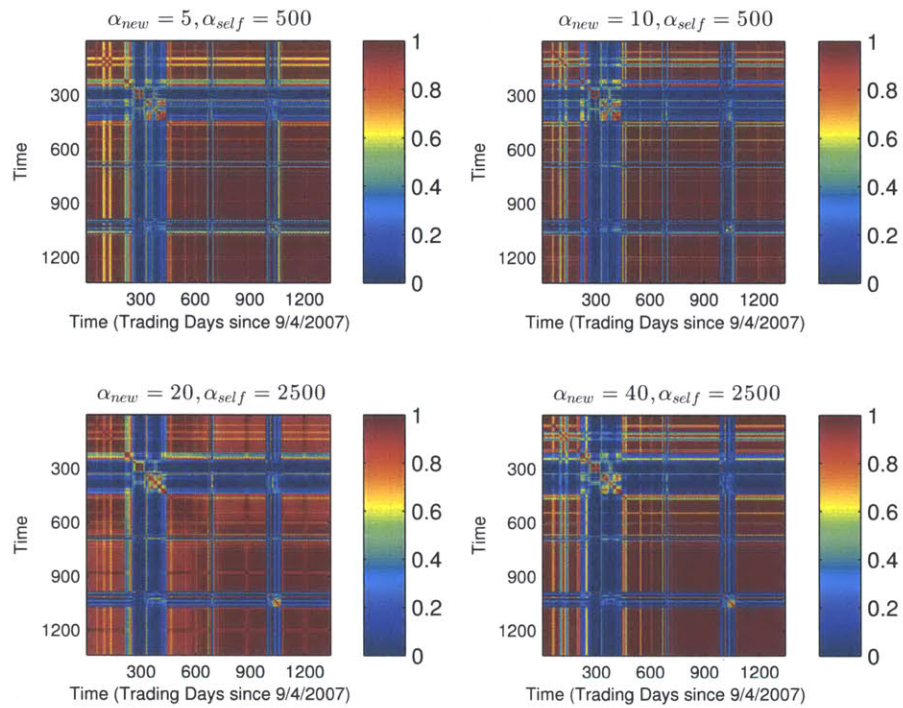
For the parameter values of  $(\alpha_{new}, \alpha_{self}) = (10, 500)$  and  $(40, 2500)$ , however, the relationship described above between XLE and XLF does not hold. Instead, the probabilities of edges are higher across the board. One explanation of the difference is the fact that more states are instantiated for these parameter values. Consequently, the number of data points assigned to each interaction structure and parameter model is smaller, which makes it easier for ONSTIM to overcome the regularization induced by the prior weighted against larger parent sets, due to the greater explanatory power of larger parent sets tailored to a small set of data points.



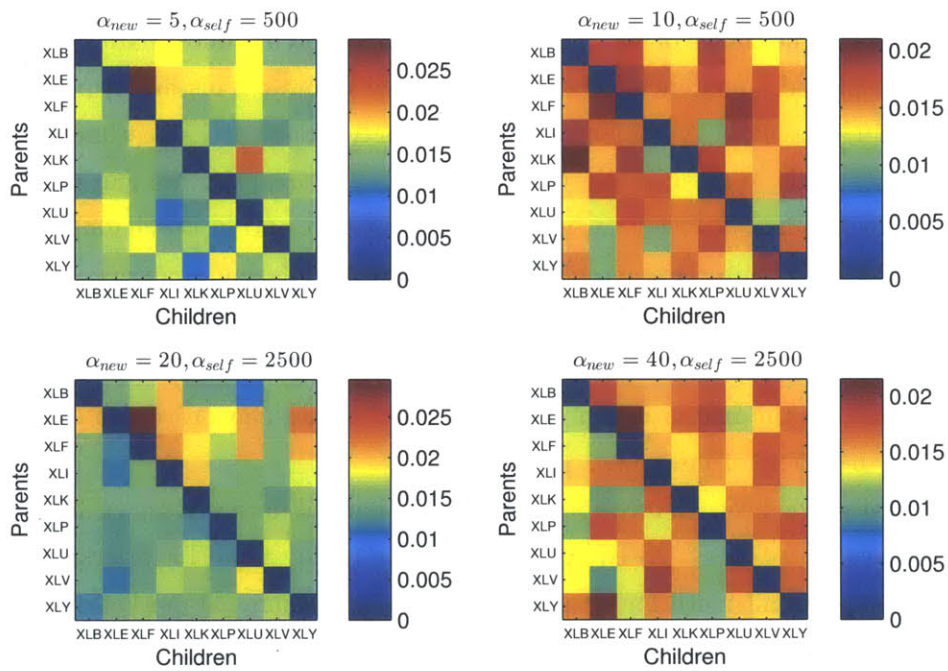
**Figure 4.13.** S&P Sector ETFs: New state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . Note that the value of the graph at time  $t$  is not necessarily the probability of a new state instantiation at time  $t$ , but rather the probability of the event that the *batch* in which  $t$  appeared instantiated a new state.



**Figure 4.14.** S&P Sector ETFs: The total instantiated numbers of states state arrival probabilities are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . The average number of inferred states for the sector datasets show significantly more sensitivity to the specified parameter values than for the Flash Crash dataset, perhaps implying that the signal in this dataset is weaker.



**Figure 4.15.** S&P Sector ETFs: Switching similarity matrices are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . The switching patterns look very similar to those shown for S&P100 stocks in Figure 4.8.



**Figure 4.16.** S&P Sector ETFs: Posteriors of possible edges are shown for the specified parameter values of  $(\alpha_{new}, \alpha_{self})$ . The value of the matrix entry  $(i, j)$  denotes the posterior probability that  $i$  is a parent of  $j$ . Posteriors for all self-edges are 1 by assumption and are thus omitted in order to improve readability for the rest of the figure.

### ■ 4.3 Summary

In this chapter, we have presented empirical results of the performance of ONSTIM on both synthetic datasets and financial datasets. ONSTIM was demonstrably able to recognize the arrival of new states both in synthetic datasets and real datasets, providing validation of our original goal of developing an online inference algorithm capable of learning model complexity from the data.

However, the performance of ONSTIM also displayed significant sensitivity to parameters such as the choice of  $\alpha_{new}$ ,  $\alpha_{self}$ , and batch offset. For most parameter settings, ONSTIM displayed a tendency to be overly aggressive towards instantiating new states, yielding estimators  $\hat{K}$  of  $K$  that typically had positive bias, and sometimes quite large positive bias. This was especially true for the case of experiments run with a batch offset of 2, in which bias was never negative or even zero, and in which global bias went as high as 20 for certain parameter settings. The accumulation of an excessive number of states is detrimental not only to accuracy of inference, but also to the efficiency of inference. For each batch, the runtime of the Gibbs sampling is linear in the number of states present, so the instantiation of small “junk” states can prove to be a significant hamper on the speed of inference.

ONSTIM yielded reasonably good results on financial datasets drawn from interday US equity prices, intraday US equity prices, and interday US ETF prices. Since there is typically no ground truth for interaction structure in financial markets, it is difficult to objectively assess the performance of ONSTIM on these datasets. However, ONSTIM successfully recognized the Flash Crash of May 6, 2010, across a variety of parameter settings with very high confidence. Further experimental work may include running experiments that extend the order of the state space model to allow for modeling of longer term dependencies between instrument prices.



# Conclusion

We began this thesis with an introduction to the problem of inferring time-varying interaction structures over a set of objects from noisy observations. We then posited the problem of designing an algorithm capable of learning these interaction structures nonparametrically, that is, without prior specification of the number of interaction structures present in the observed dataset. We further specified that the algorithm should be able to perform online inference, meaning that it should be able to learn from observations as they arrive.

In this thesis, we successfully proposed an algorithm to accomplish that goal. Online nonparametric switching temporal interaction model inference, or ONSTIM, proposes new states by sampling state indices with a probability proportional to their likelihood given the data and their prior probability given the previous state. If the sampled state index corresponds to the prior over structures and parameters, a new interaction model is instantiated. ONSTIM is capable of performing online inference by sampling state indices from this distribution as a batch of observations arrives, and then immediately instantiating a new state if necessary. By dynamically increasing the number of interaction models available to describe new observations, ONSTIM learns interaction structures nonparametrically.

We then tested ONSTIM empirically by running experiments on synthetic and financial datasets. Results from experiments on synthetic datasets were used to characterize the behavior of ONSTIM in a variety of parameter regimes, as these experiments allowed for comparison between the known true values of variables and the values that were inferred from ONSTIM. Results from financial datasets suggested the ability of ONSTIM to recognize new states.

## ■ 5.1 Drawbacks

Despite the success of ONSTIM at achieving our stated goal, there are several practical drawbacks of ONSTIM that detract from its efficiency and accuracy. First of all, as discussed in the previous chapter, ONSTIM exhibits undesirably strong sensitivity to many of the procedural parameters, especially to the batch offset. Strong variability with respect to a procedural parameter weakens the appeal of a nonparametric approach, as it effectively forces a user to optimize over another parameter instead of the original one. Instead of truly learning model complexity, some of the variability in the parameter that we wished to eliminate has just been shuffled around to other new parameters. Significant additional work needs to be undertaken to both analytically and empirically characterize the likelihood of new state instantiation as a function of the batch offset. A deeper understanding of the chain of causality between the choice of batch offset and new state instantiation is likely to be instrumental in modifying ONSTIM to alleviate such sensitivity.

As we saw in Chapter 4, the majority of parameter settings lead ONSTIM to instantiate too many switching states over the course of the inference procedure. Not only does the bloating of the procedure reduce accuracy, but every additional state also introduces a cost in terms of runtime. If the number of states instantiated is roughly linear with time, then the ONSTIM inference procedure, which is directly linear in both the number of time points and in the number of states at any time during the inference procedure, effectively has quadratic time complexity in the number of time points, a very undesirable feature.

## ■ 5.2 Further Work

The first additional steps taken would be consideration of the two problems mentioned above. Achieving a better understanding of the role of the batch offset in inference is critical, as the ability of ONSTIM to incorporate observations incrementally is entirely dependent on the batch framework.

A potential solution to the problem of too many available interaction models is to propose deletion or merges of existing interaction models alongside the instantiation of new ones. Such an approach is inspired by the split-merge Monte Carlo methods described by Hughes et al. [3], in which they develop techniques for instantiating and

merging modes in a manner so as to satisfy detailed balance.

Another area for further work is improvement of the generative process for synthetic datasets. Recall from Chapter 4 that runaway exponential growth was a common occurrence in some synthetically generated datasets, which we then discarded and resampled. Discarding samples on the basis of a specific criterion introduces bias, thereby reducing the accuracy of inference. Runaway exponential growth occurs when the transition matrix of an interaction model has eigenvalues with norm greater than 1. Since the matrix-normal prior places positive probability on all matrices, no choice of prior hyperparameters can prevent the sampling of such pathological transition matrices with certainty. A possible work-around could be the development of a sampling technique that restricts samples taken from the prior over structures and parameters to the set of stable matrices without affecting the marginal distributions of any other variables.

### ■ 5.3 Concluding Remarks

This thesis marks a step towards the integration of nonparametric approaches to model order selection into the problem of graphical model structure learning, bringing together two traditionally separate fields of inference. Despite the fact that our generative model does not fit a standard Dirichlet process-based formulation, it represents what is fundamentally a very simple process. A stochastic process that typically recurs to its previous state, sometimes revisits old states, and sometimes jumps to new states is a very commonplace notion that can model such diverse concepts as interaction structures and consumer behavior.

In some senses, the generative model that we have described lies at the heart of ONSTIM. If one focuses on the switching sequence alone, the interaction structures, latent sequence, and observed sequence can be abstracted away as just one large likelihood function to determine the posterior distribution of the switching sequence. In fact, this is exactly what the Gibbs sampler does when computing the posterior over the switching sequence  $Z$ . With that, we leave it as an exercise to the reader to determine what likelihood function to couple with the generative model next.



---

---

## Bibliography

- [1] Z. Dzunic and J. W. Fisher. Bayesian switching interaction analysis under uncertainty. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2014.
- [2] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.
- [3] M.C. Hughes, E.B. Fox, and E.B. Sudderth. Effective Split-Merge Monte Carlo Methods for Nonparametric Models of Sequential Data. In *Neural Information Processing Systems 25*. MIT Press, 2013.
- [4] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [5] 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.
- [6] SEC and CFTC. Preliminary findings regarding the market events of may 6, 2010. Technical report, SEC, CFTC, 2010.
- [7] M. R. Siracusa. *Dynamic Dependence Analysis: Modeling and Inference of Changing Dependence Among Multiple Time-Series*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 2009.
- [8] Michael R. Siracusa and John W. Fisher III. Tractable bayesian inference of time-

- series dependence structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.
- [9] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. Technical Report MIT//LIDS P-2551, MIT, Laboratory for Information and Decision Systems, 2002.
- [10] Howard E. Taylor and Samuel Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 3rd edition, February 1998.