Bayesian Nonparametric Learning of Complex Dynamical Phenomena

by

Emily B. Fox

S.B., Electrical Engineering, Massachusetts Institute of Technology, 2004
M.Eng., Elect. Eng. and Comp. Sci., Massachusetts Institute of Technology, 2005
E.E., Electrical Engineering, Massachusetts Institute of Technology, 2008

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

Doctor of Philosophy
in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

September 2009

Signature of Author: ____
Department of Electrical Engineering and Computer Science
July 31, 2009

Certified by: ____
Alan S. Willsky
Edwin Sibley Webster Professor of Electrical Engineering and Computer Science
Thesis Co-Supervisor

Certified by ____
John W. Fisher III
Principal Research Scientist
Thesis Co-Supervisor

Accepted by: ____
Terry P. Orlando
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students

Bayesian Nonparametric Learning of Complex Dynamical Phenomena

by Emily B. Fox

To be submitted to the Department of Electrical Engineering
and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The complexity of many dynamical phenomena precludes the use of linear models for which exact analytic techniques are available. However, inference on standard nonlinear models quickly becomes intractable. In some cases, *Markov switching processes*, with switches between a set of simpler models, are employed to describe the observed dynamics. Such models typically rely on pre-specifying the number of Markov modes. In this thesis, we instead take a Bayesian nonparametric approach in defining a prior on the model parameters that allows for flexibility in the complexity of the learned model and for development of efficient inference algorithms.

We start by considering dynamical phenomena that can be well-modeled as a hidden discrete Markov process, but in which there is uncertainty about the cardinality of the state space. The standard finite state *hidden Markov model* (HMM) has been widely applied in speech recognition, digital communications, and bioinformatics, amongst other fields. Through the use of the *hierarchical Dirichlet process* (HDP), one can examine an HMM with an unbounded number of possible states. We revisit this HDP-HMM and develop a generalization of the model, the *sticky HDP-HMM*, that allows more robust learning of smoothly varying state dynamics through a learned bias towards self-transitions. We show that this sticky HDP-HMM not only better segments data according to the underlying state sequence, but also improves the predictive performance of the learned model. Additionally, the sticky HDP-HMM enables learning more complex, multimodal emission distributions. We demonstrate the utility of the sticky HDP-HMM on the NIST speaker diarization database, segmenting audio files into speaker labels while simultaneously identifying the number of speakers present.

Although the HDP-HMM and its sticky extension are very flexible time series models, they make a strong Markovian assumption that observations are conditionally independent given the discrete HMM state. This assumption is often insufficient for capturing the temporal dependencies of the observations in real data. To address this issue, we develop extensions of the sticky HDP-HMM for learning two classes of switching dynamical processes: the *switching linear dynamical system* (SLDS) and the *switching vector autoregressive* (SVAR) process. These conditionally linear dynamical models can describe a wide range of complex dynamical phenomena from the stochastic volatility of financial time series to the dance of honey bees, two examples we use to show the power and flexibility of our Bayesian nonparametric approach. For all of the presented models, we develop efficient Gibbs sampling algorithms employing a truncated approximation to the HDP that allows incorporation of dynamic programming techniques, greatly improving mixing rates.

In many applications, one would like to discover and model dynamical behaviors which are shared among several related time series. By jointly modeling such sequences, we may more robustly estimate representative dynamic models, and also uncover interesting relationships among activities. In the latter part of this thesis, we consider a Bayesian nonparametric approach to this problem by harnessing the *beta process* to allow each time series to have infinitely many potential behaviors, while encouraging sharing of behaviors amongst the time series. For this model, we develop an efficient and exact Markov chain Monte Carlo (MCMC) inference algorithm. In particular, we exploit the finite dynamical system induced by a fixed set of behaviors to efficiently compute acceptance probabilities, and reversible jump birth and death proposals to explore new behaviors. We present results on unsupervised segmentation of data from the CMU motion capture database.

Thesis Supervisors: Alan S. Willsky
Professor of Electrical Engineering and Computer Science

John W. Fisher III
Principal Research Scientist

Acknowledgments

Everything should be made as simple as possible, but not simpler.
attributed to Albert Einstein

Aerodynamically the bumblebee shouldn't be able to fly,
but the bumblebee doesn't know that so it goes on flying anyway.
Mary Kay Ash

This thesis marks the culmination of an intense though incredibly gratifying journey at MIT that started nearly a decade ago. I look fondly upon my years as an undergraduate student at the Institution, but it was my time as a graduate student that was the most formative and rewarding. Academically, this is in large part due to the interactions I had with my advisor, Professor Alan Willsky. Alan's incredible breadth and depth of knowledge have been an inspiration to me and of great importance in shaping the research contained in this thesis. No matter how many times I ventured away from the group's core areas, Alan was always right there still actively (and energetically!) following and providing context for the ideas. My co-advisor, Dr. John Fisher, has provided, in addition to many good laughs and distractions from work, illumination into my research through many insightful questions; he was also readily available to answer all of my many questions.

In addition to my interactions with Alan and John, the other students in the Stochastic Systems Group (SSG), both past and present, have played a pivotal role in my graduate studies. My long-time officemates—Kush Varshney, Pat Kreidl, and Jason Williams—provided stimulating conversations and tolerated my incessant interruptions. My new officemate, Matt Johnson, has quickly filled those shoes since Pat and Jason graduated. We have had many interesting discussions on Bayesian statistics and I look forward to continued collaborations. I also want to thank Myung Jin Choi, Venkat Chandrasekaran, Vincent Tan, and Ying Liu for enlivening SSG with Friday poker night and other group events. Along those lines, I thank members of CSAIL, such as Mike Siracusa, Gerald Dalley, Wanmei Ou, and Thomas Yeo, for sharing in "vulturing" trips and our ensuing lunch conversations. I am particularly indebted to Mike Siracusa who, in addition to many illuminating discussions on Markov switching processes, went above and beyond in helping me with computing issues. We made a fabulous, and seemingly automatic, grouplet and SSG seminar pairing.

I must highlight Erik Sudderth, a former SSG member, for the exceptional guidance and mentoring he has provided me during my graduate studies. I can attribute my

exposure to Bayesian nonparametrics to Erik, who examined such methods during the latter part of his thesis. Although our collaborations started only after he left MIT, Erik has contributed significantly to the work presented herein. He has taught me a great deal about persistence (no pun intended regarding Chapter 3) and thorough analysis of results. My thesis committee, Professor Munzer Dahleh and Princeton University's Professor David Blei, also provided thoughtful suggestions that continue to guide my research and help make the work understandable to both the System Identification and Machine Learning communities. In addition to Alan's courses on recursive estimation and stochastic systems, Munzer's exceptional instruction of the course on dynamic systems and control was pivotal in building my foundation for studying the dynamical models presented in this thesis.

I have also had the honor of working with Professor Michael Jordan at UC Berkeley. During my many visits to Berkeley, and through a massive number of emails, this bicoastal collaboration has provided me with invaluable guidance on my work in Bayesian nonparametrics and insight into the fields of Machine Learning and Statistics. Additionally, I am deeply indebted to Mike for his extensive editing of papers that comprise a good portion of this thesis. Another contributing factor outside of MIT's campus was my time interning at MIT Lincoln Laboratory, specifically working on target tracking, that set me on this hunt for flexible descriptions of Markov switching models. Without the inspiration of that application, and discussions with Keh Ping Duhn, David Choi, and Daniel Rudoy, I likely would not have taken the path I did.

On a more personal note, I would like to thank my family for their support during my nine-year adventure 3,000 miles away from home. I would especially like to acknowledge my mom who has always supported my pursuits, however offbeat and incomprehensible they were to her (e.g., ice hockey, pole vaulting, and needless to say, anything having to do with math.) My stepdad, who has a Ph.D. in chemistry, has been a refuge at home in understanding logical reasoning while my dad has taught me the value of adventure and optimism. My siblings, Ben and Nathan, have each been there for me in incredible ways. I also must thank all of my friends, especially Melanie Rudoy[1] and Erin Aylward, for their never-ending support, encouragement, and distractions. Finally, this endeavor would have been infinitely more challenging without the love and support of Wes McKinney[2].

[1] Yes, there are two Rudoys in one acknowledgments section.

[2] For him, I must thank Jim Munkres who taught the topology course in which we met.

Contents

Abstract 3

Acknowledgments 5

List of Figures 11

List of Algorithms 15

List of Tables 17

Notational Conventions 19

1 Introduction 23
- 1.1 Thesis Organization and Overview of Methods and Contributions 26
  - 1.1.1 Chapter 2: Background 27
  - 1.1.2 Chapter 3: The Sticky HDP-HMM 27
  - 1.1.3 Chapter 4: Bayesian Nonparametric Learning of SLDS 28
  - 1.1.4 Chapter 5: Sharing Features among Dynamical Systems with Beta Processes 30
  - 1.1.5 Chapter 6: Contributions and Recommendations 31
  - 1.1.6 Appendices 31

2 Background 33
- 2.1 The Bayesian Framework 33
  - 2.1.1 Modeling via Exchangeability 34
- 2.2 Exponential Families 40
  - 2.2.1 Properties of the Canonical Exponential Family 42
  - 2.2.2 Interpretation as Linearly Constrained Maximum Entropy Distribution 44
  - 2.2.3 Examples 46
- 2.3 Sufficient Statistics 47
- 2.4 Incorporating Prior Knowledge 49

2.4.1 Conjugate Priors 50
2.4.2 Multinomial Observations 52
2.4.3 Gaussian Observations 53
2.4.4 Multivariate Linear Regression Model 55
2.5 Graphical Models 57
2.5.1 A Brief Overview 57
2.5.2 Directed Graphical Models 58
2.5.3 Undirected Graphical Models 60
2.5.4 Belief Propagation 62
2.6 Hidden Markov Model 66
2.6.1 Forward-Backward Algorithm 67
2.6.2 Viterbi Algorithm 69
2.7 State Space Models 71
2.7.1 Standard Discrete-Time Linear-Gaussian State Space Formulation 71
2.7.2 Vector Autoregressive Processes 72
2.7.3 Switching Linear Dynamic Systems 72
2.7.4 Stochastic Realization Theory 73
2.7.5 Kalman Filtering and Smoothing 76
2.8 Markov Chain Monte Carlo 80
2.8.1 Monte Carlo Integration 80
2.8.2 The Metropolis-Hastings Algorithm 81
2.8.3 Gibbs Sampling 83
2.8.4 Auxiliary, Blocked, and Collapsed Gibbs Samplers 86
2.9 Bayesian Nonparametric Methods 91
2.9.1 Dirichlet Processes 92
2.9.2 Dirichlet Process Mixture Models 96
2.9.3 Hierarchical Dirichlet Processes 98
2.9.4 Beta Process 102

3 The Sticky HDP-HMM 107

3.1 The HDP-HMM and Its Sticky Extension 109
3.1.1 Chinese Restaurant Franchise with Loyal Customers 111
3.1.2 Sampling via Direct Assignments 114
3.1.3 Blocked Sampling of State Sequences 115
3.1.4 Hyperparameters 117
3.2 Experiments with Synthetic Data 117
3.2.1 Gaussian Emissions 119
3.2.2 Multinomial Emissions 124
3.2.3 Comparison to Independent Sparse Dirichlet Prior 125
3.3 Multimodal Emission Densities 126
3.3.1 Direct Assignment Sampler 127
3.3.2 Blocked Sampler 128

3.4 Assessing the Multimodal Emissions Model 128
3.4.1 Mixture of Gaussian Emissions 128
3.5 Speaker Diarization 132
3.6 Discussion and Future Work 137

4 Bayesian Nonparametric Learning of SLDS 141
4.1 The HDP-SLDS and HDP-AR-HMM Models 143
4.1.1 Posterior Inference of Dynamic Parameters 145
Conjugate Prior on $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$ 147
Alternative Prior — Automatic Relevance Determination 147
Measurement Noise Posterior 152
4.1.2 Gibbs Sampler 153
Sampling Dynamic Parameters $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$ 153
Sampling Measurement Noise R (HDP-SLDS only) 154
Block Sampling $z_{1:T}$ 154
Block Sampling $\boldsymbol{x}_{1:T}$ (HDP-SLDS only) 154
Sequentially Sampling $z_{1:T}$ (HDP-SLDS only) 155
4.2 Results 156
4.2.1 MNIW prior 156
4.2.2 ARD prior 160
4.2.3 Dancing Honey Bees 163
4.3 Model Variants 169
4.3.1 Shared Dynamic Matrix, Switching Driving Noise 169
4.3.2 Fixed Dynamic Matrix, Switching Driving Noise 174
4.4 Discussion and Future Work 181

5 Sharing Features among Dynamical Systems with Beta Processes 183
5.1 Describing Multiple Time Series with Beta Processes 184
5.2 MCMC Methods for Posterior Inference 186
5.2.1 Sampling binary feature assignments 187
5.2.2 Sampling dynamic parameters and transition variables 191
5.2.3 Sampling the IBP and Dirichlet transition hyperparameters . . . 192
5.3 Synthetic Experiments 194
5.4 Motion Capture Experiments 198
5.5 Discussion and Future Work 203

6 Contributions and Recommendations 205
6.1 Summary of Methods and Contributions 205
6.2 Suggestions for Future Research 207
6.2.1 Inference on Large-Scale Data 207
6.2.2 Alternative Dynamic Structures 208
6.2.3 Bayesian Nonparametric Variable-Order Markov Models 209
6.2.4 Alternatives to Global Clustering 210

6.2.5 Asymptotic Analysis 210

A Sticky HDP-HMM Direct Assignment Sampler **213**

A.1 Sticky HDP-HMM 213

A.1.1 Sampling z_t 213

A.1.2 Sampling β 217

A.1.3 Jointly Sampling m_{jk}, w_{jt}, and $\bar{m}_{jk}$ 218

A.2 Sticky HDP-HMM with DP emissions 220

B Sticky HDP-HMM Blocked Sampler **223**

B.1 Sampling β, π, and ψ 223

B.2 Sampling $z_{1:T}$ for the Sticky HDP-HMM 224

B.3 Sampling $(z_{1:T}, s_{1:T})$ for the Sticky HDP-HMM with DP emissions . . . 224

B.4 Sampling θ 225

B.4.1 Non-Conjugate Base Measures 225

C Hyperparameters **227**

C.1 Posterior of $(\alpha + \kappa)$ 227

C.2 Posterior of γ 229

C.3 Posterior of σ 230

C.4 Posterior of ρ 231

D HDP-SLDS and HDP-AR-HMM Message Passing **233**

D.1 Mode Sequence Message Passing for Blocked Sampling 233

D.2 State Sequence Message Passing for Blocked Sampling 234

D.3 Mode Sequence Message Passing for Sequential Sampling 237

E Derivation of Maneuvering Target Tracking Sampler **243**

E.1 Chinese Restaurant Franchise 244

E.2 Normal-Inverse-Wishart Posterior Update 244

E.3 Marginalization by Message Passing 245

E.4 Combining Messages 245

E.5 Joining Distributions that Depend on $\boldsymbol{u}_t$ 248

E.6 Resulting $(\boldsymbol{u}_t, z_t)$ Sampling Distributions 248

F Dynamic Parameter Posteriors **251**

F.1 Conjugate Prior — MNIW 251

F.2 Non-Conjugate Independent Priors on $\mathbf{A}^{(k)}$, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$ 254

F.2.1 Normal Prior on $\mathbf{A}^{(k)}$ 254

F.2.2 Inverse Wishart Prior on $\Sigma^{(k)}$ 255

F.2.3 Normal Prior on $\boldsymbol{\mu}^{(k)}$ 255

Bibliography **257**

List of Figures

1.1 Examples of data we examine in the thesis. 25
1.2 Motion capture skeleton plots for six examples of jumping jacks. 26

2.1 Histograms of inferred parameters from coin flipping and Pólya urn experiments. 38
2.2 Bayes ball algorithm. 59
2.3 Graphical representation of Markov blanket. 60
2.4 Hierarchical Bayesian model of exchangeable random variables. 60
2.5 Moralization of two directed graphical models. 61
2.6 Example tree graphical models. 63
2.7 Graphical representation of two belief propagation scheduling schemes. . 65
2.8 Graphical representation of a hidden Markov model (HMM). 66
2.9 Lattice representation of an HMM state sequence. 67
2.10 Graphical models for the switching vector autoregressive (VAR) process and switching linear dynamical system (SLDS). 74
2.11 Graphical model of a finite mixture model. 88
2.12 Dirichlet process mixture model graphs. 97
2.13 Graphical model of Chinese restaurant franchise. 100
2.14 Depiction of a Chinese restaurant franchise with two restaurants. 101
2.15 A draw from a beta process, and associated Bernoulli realizations, along with a realization from the Indian buffet process. 103

3.1 Demonstration of rapid transitions in HDP-HMM state sequences. . . . 108
3.2 Sticky HDP-HMM graphical models. 110
3.3 Graphical model of Chinese restaurant franchise with loyal customers. . 112
3.4 Illustration of dish-choosing process for the Chinese restaurant with loyal customers. 114
3.5 Demonstration of sequential HDP-HMM Gibbs sampler splitting temporally separated examples of the same state. 115
3.6 Synthetic three-state HMM observation sequence and resulting sticky vs. non-sticky HDP-HMM performance. 119

3.7 Comparison of perfromance of the blocked and sequential HDP-HMM Gibbs samplers on the three-state HMM synthetic observation sequence. 120
3.8 Performance of beam sampling on the three-state HMM synthetic data example. 121
3.9 Fast state-switching synthetic data along with sticky vs. non-sticky HDP-HMM segmentation performance. 123
3.10 Multinomial synthetic data along with sticky vs. non-sticky HDP-HMM segmentation results. 125
3.11 State transition diagram for a nine-state HMM. 126
3.12 Sticky HDP-HMM results for the nine-state HMM example, as compared to a model with an independent sparse Dirichlet prior. 127
3.13 Sticky vs. non-sticky HDP-HMM performance on data generated from a five-state HMM with mixture of Gaussian emissions. 131
3.14 Block diagram of preprocessing of speaker diaraization data. 133
3.15 For each of the 21 meetings, comparison of diarizations using sticky vs. original HDP-HMM with DP emissions. 134
3.16 Example diarization for the NIST_20051102-1323 meeting. 135
3.17 Example diarization for the VT_20050304-1300 meeting. 137
3.18 Chart comparing the DERs of the sticky and original HDP-HMM with DP emissions to those of ICSI for each of the 21 meetings. 138
3.19 Trace plots of log-likelihood and Hamming distance error for 10 chains over 100,000 Gibbs iterations for the NIST_20051102-1323 meeting. . . . 139

4.1 Graphical models of the HDP-SLDS and an order two HDP-AR-HMM. 144
4.2 Block diagram of one iteration of the Gibbs sampler for the HDP-SLDS and HDP-AR-HMM. 146
4.3 Depiction of HDP-SLDS sampling stages on a set of graphical models. . 153
4.4 Depiction of HDP-AR-HMM sampling stages on a set of graphical models 154
4.5 Plots of three synthetic data sequences generated from switching linear dynamical processes. 161
4.6 Synthetic data results for the three sequences using the HDP-SLDS, HDP-AR-HMM, and HDP-HMM. 161
4.7 Synthetic data generated from an SLDS with a sparse dynamical matrix, and results comparing the HDP-SLDS with an ARD vs. MNIW prior. . 163
4.8 Trajectories of six honey bees dance sequences. 164
4.9 Change-point detection performance of the HDP-AR-HMM on the six honey bee dance sequences as compared to the method of Xuan and Murphy [188]. 165
4.10 Segmentation performance of the HDP-AR-HMM on the six honey bee dance sequences. 166
4.11 Honey bee head angle measurements for the six dances. 167
4.12 Inferred ARD hyperparameters for the learned honey bee dance modes. 168

4.13 IBOVESPA stock index daily returns from 01/03/1997 to 01/16/2001. . 170
4.14 IBOVESPA stock index change-point detection performance for two variants of the HDP-SLDS. 173
4.15 HDP-SLDS maneuvering target tracking results for synthetic data of a target following a sinusoidal trajectory. 180
4.16 HDP-SLDS maneuvering target tracking results for synthetic data of a target controlled by a step function input on acceleration. 181

5.1 Graphical model of the IBP-AR-HMM. 186
5.2 Synthetic data for 5 switching AR(1) time series, and associated true and IBP-AR-HMM learned feature matrices. 197
5.3 Hamming distance quantiles comparing the segmentation performance of the HDP-AR-HMM to the IBP-AR-HMM on a synthetic data example. 199
5.4 Motion capture skeleton plots for IBP-AR-HMM learned segmentations of six exercise routine videos. 200
5.5 Comparison of the IBP-AR-HMM MoCap segmentation performance to HMM and Gaussian mixture model approaches. 201
5.6 Learned MoCap feature matrices from the IBP-AR-HMM, HMM, and Gaussian mixture model approaches. 202

List of Algorithms

1 Viterbi hidden Markov model decoding. 70

2 Kalman filter recursion for an LTI system. 76

3 Stable forward information form Kalman filter recursion. 79

4 Metropolis-Hastings algorithm. 81

5 Multi-stage Gibbs sampling algorithm. 84

6 Two-stage Gibbs sampling algorithm. 86

7 Completion Gibbs sampler for a finite mixture model. 90

8 Collapsed Gibbs sampler for a finite mixture model. 91

9 Direct assignment collapsed Gibbs sampler for the sticky HDP-HMM. . 116

10 Blocked Gibbs sampler for the sticky HDP-HMM. 118

11 Direct assignment collapsed Gibbs sampler for the sticky HDP-HMM with DP emissions. 129

12 Blocked Gibbs sampler for the sticky HDP-HMM with DP emissions. . . 130

13 HDP-SLDS and HDP-AR-HMM Gibbs sampler. 157

14 Blocked mode-sequence sampler for HDP-AR-HMM or HDP-SLDS. . . 158

15 Parameter sampling using MNIW prior. 158

16 Parameter sampling using ARD prior. 159

17 IBP-AR-HMM MCMC sampler. 195

18 IBP-AR-HMM auxiliary variable sampler for updating transition and dynamic parameters. 196

19 Numerically stable form of the backwards Kalman information filter. . . 237

20 Numerically stable form of the forward Kalman information filter. . . . 239

List of Tables

3.1 Overall DERs for the sticky and original HDP-HMM with DP emissions. 136

4.1 Median label accuracy of the HDP-AR-HMM compared to accuracy of the approach of Oh et al. [129]. 167
4.2 Table of 10 key world events affecting the IBOVESPA stock index from 01/03/1997 to 01/16/2001. 171
4.3 Summary of two variants of the HDP-SLDS for detecting changes in volatility of a stock index. 172

Notational Conventions

Symbol	Definition
General Notation	
$\mathbb{Z}_+$	the set of positive integers
$\mathbb{R}$	the set of reals
$x_{1:t}$	the sequence $\{x_1, \ldots, x_t\}$
$x_{\setminus t}$	the sequence $\{x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T\}$, where T is largest possible index
$x_{\cdot b}$	$\sum_a x_{ab}$
$x_{a \cdot}$	$\sum_b x_{ab}$
$x_{\cdot\cdot}$	$\sum_b \sum_a x_{ab}$
$\lvert \cdot \rvert$	cardinality of a set
$\delta(k, j)$	the discrete Kronecker delta
δ_θ	measure concentrated at θ
$\mathbb{E}[\cdot]$	expectation of a random variable
$\mathrm{DP}(\alpha, H)$	Dirichlet process distribution with concentration parameter α and base measure H
$\mathrm{Dir}(\alpha_1, \ldots, \alpha_K)$	K-dimensional finite Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_K$
$\mathrm{Ber}(p)$	Bernoulli distribution with parameter p
$\mathrm{GEM}(\gamma)$	stick-breaking distribution with parameter γ

Symbol	Definition
Hierarchical Dirichlet Process and Chinese Restaurant Franchise with Loyal Customers	
y_{ji}	i^{th} observation within j^{th} group
z_{ji}	index of mixture component that generated observation y_{ji}
θ_{ji}	(non-unique) parameter associated with observation y_{ji}
θ^*_{jt}	(non-unique) parameter, or *dish*, served at table t in restaurant j
θ^{**}_k	k^{th} unique global parameter of the mixture model
t_{ji}	table assignment for observation, or *customer*, y_{ji}
$\bar{k}_{jt}$	considered dish assignment for table t in restaurant j
k_{jt}	served dish assignment for table t in restaurant j
$\underline{\bar{k}}_j$	the set of all considered dish assignments in restaurant j
$\underline{k}_j$	the set of all served dish assignments in restaurant j
w_{jt}	override variable for table t in restaurant j
$\tilde{n}_{jt}$	number of customers at table t in restaurant j
$\bar{m}_{jk}$	number of tables in restaurant j that considered dish k
m_{jk}	number of tables in restaurant j that were served dish k
T_j	number of currently occupied tables in restaurant j
$\bar{K}$	number of unique dishes considered in the franchise
K	number of unique dishes served in the franchise
Sticky HDP-HMM	
y_t	observation from the hidden Markov model at time t
z_t	state of the Markov chain at time t
n_{jk}	number of transitions from state j to state k in $z_{1:T}$
n^{-t}_{jk}	number of transitions from state j to state k in $z_{1:T}$, not counting the transitions $z_{t-1} \rightarrow z_t$ or $z_t \rightarrow z_{t+1}$
κ	self-transition parameter
ρ	self-transition proportion parameter $\kappa/(\alpha+\kappa)$
with DP emissions	
s_t	index of mixture component that generated observation y_t
n'_{jk}	number of transitions from state j to state k in $z_{1:T}$
n^{-t}_{jk}	number of transitions from state j to state k in $z_{1:T}$, not counting the transitions $z_{t-1} \rightarrow z_t$ or $z_t \rightarrow z_{t+1}$
K'_j	number of currently instantiated mixture components for state j's emission distribution

Symbol	Definition
HDP-SLDS and HDP-AR-HMM	
VAR(r)	order r vector autoregressive process
SLDS	switching linear dynamical system
$A_i^{(k)}$	i^{th} lag matrix of the k^{th} VAR process
$\mathbf{A}^{(k)}$	dynamic matrix for k^{th} dynamical mode For HDP-AR-HMM, contains lag matrices $A_i^{(k)}$
$\Sigma^{(k)}$	process noise covariance of k^{th} dynamical mode
C	measurement matrix
R	measurement noise covariance
z_t	dynamical mode index at time t
$\boldsymbol{x}_t$	continuous-valued state vector at time t
$\boldsymbol{y}_t$	observation vector at time t
$\boldsymbol{\psi}_t$	pseudo-observation vector at time t
$\bar{\boldsymbol{\psi}}_t$	lag pseudo-observation vector at time t
d	dimension of the observations $\boldsymbol{y}_t$
n	dimension of the latent state $\boldsymbol{x}_t$
$\mathcal{S}_\ell$	set of indices for which elements $a_{ij}^{(k)}$ of $\mathbf{A}^{(k)}$ are distributed with ARD parameter $\alpha_\ell^{(k)}$

Symbol	Definition
IBP-AR-HMM	
$z_t^{(i)}$	dynamical mode index for object i at time t
$\mathbf{y}_t^{(i)}$	observation vector for object i at time t
$\boldsymbol{f}_i$	feature vector for object i containing elements f_{ik}
$\eta_{jk}^{(i)}$	transition variables for object i
$\boldsymbol{A}_k$	dynamic matrix for k^{th} dynamical mode
Σ_k	process noise covariance of k^{th} dynamical mode
$\pi_j^{(i)}$	j^{th} feature-constrained transition distribution for object i Normalizes $\eta_{jk}^{(i)}$ over indices determined by $\boldsymbol{f}_i$
K_+	total number of instantiated dynamical modes
K_+^{-i}	number of instantiated dynamical modes not considering those used by object i
$\boldsymbol{f}_{-i}$	feature vector for object i containing only the components of $\boldsymbol{f}_i$ shared by other objects
$\boldsymbol{f}_{+i}$	feature vector for object i containing the feature indices of $\boldsymbol{f}_i$ unique to object i
$\boldsymbol{\theta}_+$	dynamic parameters $\boldsymbol{\theta}_k = \{\boldsymbol{A}_k, \Sigma_k\}$ for features unique to object i
$\boldsymbol{\eta}_+$	transition variables $\eta_{jk}^{(i)}$ associated with features unique to object i
n_i	number of features unique to object i

Chapter 1

Introduction

THE study of dynamical phenomena is pervasive in fields as diverse as bioinformatics, econometrics, and systems and control. For example, within bioinformatics one might be interested in modeling recombination hotspots and ancestral haplotypes. In econometrics, classical time series include daily returns of a stock index, the exchange rate of a currency, or interest rate. Systems and controls applications are plentiful, ranging from robotics to modeling the dynamics of aircraft. Within these fields, there has been an explosion of data of increasingly complex phenomena, resulting in a push toward building more intricate time series models and developing efficient inference techniques. The challenges these datasets pose result from a convergence of factors: the size of the datasets demand examination of time series analysis techniques that scale effectively with the dimensionality of the data while the complexity of the dynamics precludes the use of standard linear dynamical models for which exact inference techniques exist.

A small subset of time series data, such as the trajectory of a ballistic missile, can be described by a single dynamical model that is well-defined through knowledge of the underlying physics of the object we are observing. Slightly more complicated time series, like a maneuvering passenger aircraft, can be described as switching between a small set of dynamical models. However, many of the dynamical processes we encounter are too complex for such modeling schemes. For example, describing human motion requires the formulation of a model that represents the large number of degrees of freedom provided by the many human joints. High performance aircraft or the dance of honey bees [129] are other examples of dynamical systems with patterned, but very intricate motions. In this thesis, we consider methods for learning dynamical models for time series with complex and uncertain behavior patterns. Specifically, we address how Bayesian nonparametric methods can be used to provide a flexible and computationally efficient structure for learning and inference of these complex systems.

Although the true underlying dynamics of the phenomena of interest are generally nonlinear, they can often be effectively modeled as switches among a set of conditionally linear dynamical modes. These *switching linear dynamical systems* (SLDS) have been used to describe, for example, human motion [133, 140], financial time series [27, 94, 154], and maneuvering targets [43, 145]. Within the control community, these models are often referred to as *Markov jump-linear systems* (MJLS). The different linear dynamical modes account for changes the phenomena exhibit: a person changes from walking to running; a country undergoes a recession, a central bank intervention, or some national or global event; an aircraft makes an evasive maneuver. Classical methods for inferring the latent state of the switching dynamical process rely on defining a fixed, finite set of models with known parameterizations and switching behaviors. In the case of identifying switching dynamical processes, the field consists of only a fixed number of directions: either relying on knowledge of the number of dynamical regimes and estimating the model parameters from the data, or relying on simplifying assumptions such as deterministic dynamics when the number of models is not known. Further details are discussed in Chapter 4. Alternatively, emerging methods within the field of Bayesian nonparametrics, specifically hierarchical extensions of the Dirichlet process, offer promise in learning stochastic switching dynamical models with the flexibility of incorporating new dynamical modes as new behaviors are observed. Furthermore, by casting the problem of system identification within the Bayesian framework, one can leverage the extensive theory and methodologies of this field.

The clustering properties induced by the Dirichlet process prior have been exploited in many standard mixture modeling applications. Hierarchical layerings of Dirichlet processes, such as the hierarchical Dirichlet process (HDP) [162] and the nested Dirichlet process [143], as well as generalizations of the Dirichlet process, such as the Pitman-Yor process [72, 137], have proven useful in a variety of fields including genomics [187], document modeling [19], natural language processing [58, 160], and computer vision [158]. Originally developed for static estimation problems, a burgeoning trend is realizing the significant impact these methods can have on time series analysis, an impact which cuts through the boundaries between machine learning, statistics, and dynamics and control. One perspective of this analysis has been the development of Dirichlet process priors on stochastically evolving distributions such as the dependent Dirichlet process [61, 111] and the kernel stick-breaking process [39]. For example, imagine one has recordings of a unknown collection of neurons. Due to either changing recording conditions or changes within the neuron itself, the waveforms observed may vary with time. In such cases, one would like to allow the model parameters to stochastically evolve [48]. Other uses of these processes include the study of how a response density changes with predictors [39], or time-varying document topic modeling [156] in which the popularity of various topics within a given domain evolve with time.

The complex time series we analyze in this thesis, however, have more patterned behaviors that we would like to capture through models that allow repeated returns to a set of simpler dynamical models. In such cases, instead of examining stochastically evolving distributions as in the dependent Dirichlet process, we would like to nonparametrically model the stationary transition distributions of a discrete-time Markov process. That is, we would like to allow for switching processes with an unbounded number of possible Markov states. A first attempt at such a model is the hierarchical Dirichlet process hidden Markov model (HDP-HMM) [11, 162]. One of our contributions in this thesis—the *sticky* HDP-HMM—provides improved control over the number of hid-

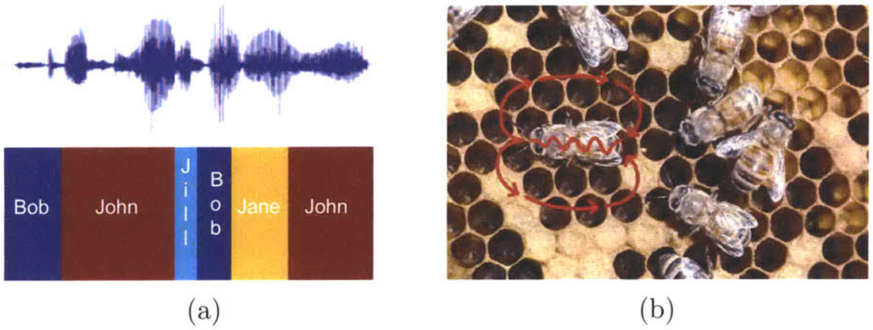


Figure 1.1. Two examples of data we examine in the thesis. (a) A speech signal from which we aim to infer the number of speakers and a segmentation of the audio into speaker labels; (b) A honey bee in the beehive, performing a set of three dances indicated by the arrows: *turn right*, *turn left*, and *waggle*. In this scenario, our goal is to discover these dances and to estimate dynamical models to describe them.

den Markov model modes inferred by better capturing the temporal mode persistence present in many real datasets. As a motivating example for the sticky HDP-HMM, consider the problem of *speaker diarization* [185], to which we return in Chapter 3. Here, an audio recording is made of a meeting involving multiple human participants and the problem is to segment the recording into time intervals associated with individual speakers. See Fig. 1.1(a). Segmentation is to be accomplished without a priori knowledge of the number of speakers involved in the meeting; moreover, one does not assume a priori knowledge of the speech patterns of particular individuals. For this application, we show that producing state-of-the-art diarizations using the HDP-HMM requires the sticky extension to properly account for the fact that a person currently speaking is likely to continue speaking.

Both the HDP-HMM and its sticky extension make a strong Markovian assumption that observations are conditionally independent given the mode. Such an assumption is inappropriate for many of the datasets we examine. For example, consider the problem of segmenting the dance of a honey bee into the *turn right*, *turn left*, and *waggle* dances depicted in Fig. 1.1(b) [129]. (See Chapter 4 for explanation.) In such a scenario, even conditioned on the dance mode, the observations of the honey bee position are highly correlated and thus the overall dance cannot be well approximated by a hidden Markov model. Motivated by such applications, in this thesis we also examine a Bayesian nonparametric approach for learning SLDS, thereby capturing a broader class of dynamical phenomena exhibiting more complex temporal dependencies.

While the Dirichlet process targets inferring a small set of representative dynamical modes, there is still a question about the dimensionality of the parametrization for the conditionally linear dynamical models. In the presence of limited data, one would like to reduce the number of parameters that must be estimated. Additionally, finding the minimal such dimension still yielding a model adequately describing the observed dynamics can provide insight into properties of the underlying dynamical phenomenon.

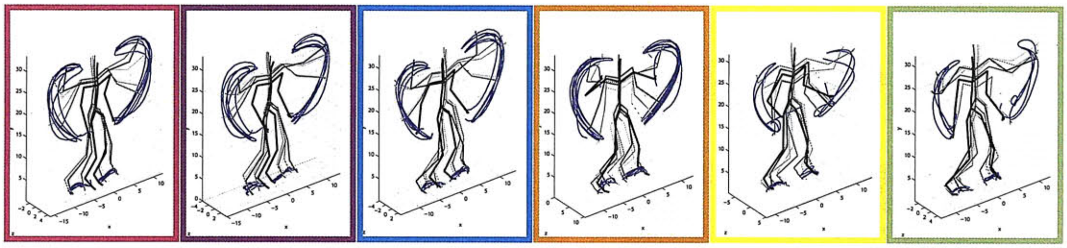


Figure 1.2. Motion capture skeleton plots for six examples of jumping jacks, each from a different motion capture movie. Skeleton rendering done by modifications to Neil Lawrence's Matlab MoCap toolbox [105].

To jointly address these issues, we propose a method of inducing sparsity in the temporal dependency structure among variables.

In the problems discussed so far, we have assumed that we are interested in the dynamics of a single time series. However, in many applications one is presented with numerous realizations of related phenomena. One example we consider in Chapter 5 is that of motion capture data depicting multiple people performing a set of related tasks. In such cases, one would like to discover and model dynamical behaviors which are shared among the multiple, related time series. For example, in the motion capture data one might be interested in grouping all instances of jumping jacks from a collection of videos, as displayed in Fig. 1.2. The benefits of such joint modeling are twofold: we may more robustly estimate representative dynamic models in the presence of limited data, and we may also uncover interesting relationships among the time series. Our proposed method relates the set of dynamical behaviors each object exhibits through utilization of a beta process prior [67, 165]. This specific choice of a Bayesian nonparametric prior allows flexibility in the number of total and sequence-specific behaviors, and encourages the time series to share similar subsets of the large set of possible behaviors.

■ 1.1 Thesis Organization and Overview of Methods and Contributions

We now provide an overview of the contributions of each chapter, including methodologies and results, as well as an overview of the chapter structure. The introductory paragraphs of each chapter provide more detailed outlines.

The overarching theme of the thesis is the proposal of methods for Bayesian nonparametric learning of time series exhibiting complex dynamics that can be approximated as switches among conditionally linear dynamical modes. For each of the Bayesian nonparametric time series models that we present, we leverage the simple Markov structure and the induced conditionally linear dynamics to develop efficient inference techniques. Throughout this thesis, we provide numerous demonstrations that our proposed Bayesian nonparametric framework provides flexible and efficient methods for learning simple representative models of dynamical phenomena from limited noisy observations.

■ 1.1.1 Chapter 2: Background

We begin by reviewing many of the statistical concepts that are utilized throughout this thesis. The chapter starts by motivating the Bayesian, and more specifically the Bayesian *nonparametric*, approach by presenting the de Finetti theorem. We then describe exponential families of probability distributions and sufficient statistics. Together, these concepts enable examination of prior distributions, namely conjugate prior distributions, that lead to efficient inference techniques built upon in this thesis. We present an analysis of a class of likelihood models and associated conjugate priors used extensively in our models. The chapter then moves to discussing the graphical model representation of joint probability distributions. We provide an introduction to graphical models, with an emphasis on the directed chains and their associated inference techniques that provide the basis for the models we consider and are fundamental to our derivations. For the more general models we consider in this thesis, exact inference is infeasible and we rely on the Markov chain Monte Carlo techniques outlined in this chapter. We conclude the chapter with background material on the stochastic processes we use in developing our Bayesian nonparametric models: the Dirichlet process, its hierarchical extension, and the beta process.

■ 1.1.2 Chapter 3: The Sticky HDP-HMM

Accounting for Temporal Mode Persistence

The existing Bayesian nonparametric approach to learning hidden Markov models (HMMs)—the HDP-HMM [162]—utilizes the hierarchical Dirichlet process (HDP) to allow for an unbounded HMM mode space. However, as we thoroughly analyze in this chapter, the HDP-HMM inadequately captures the temporal mode persistence present in many real datasets such as the speaker diarization application described previously. To address this issue, we augment the model with a bias towards self-transitions and show that in our scenarios of interest this *sticky HDP-HMM* leads to both improved segmentation performance as well as increased predictive power. Earlier papers have also proposed self-transition parameters for HMMs with infinite mode spaces [11, 186], but did not formulate general solutions that integrate fully with Bayesian nonparametric inference. One of the main contributions of this chapter is the derivation of an exact Gibbs sampling technique that allows for a *learned* bias towards self-transitions instead of relying on fixing this sticky parameter. As such, the model still allows for fast-switching dynamics if they are present in the data.

Efficient Inference Leveraging Dynamic Programming

The direct assignment Gibbs sampler [162] developed for inference in the hierarchical Dirichlet process was also proposed as the sampler for the HDP-HMM. This direct assignment sampler marginalizes the HMM transition distributions and sequentially samples the mode sequence. However, as we demonstrate in this chapter, sequential sampling of a mode sequence with strong correlations leads to very slow mixing rates.

This problem is exacerbated in the case of the sticky HDP-HMM in which the temporal dependencies encoded in the prior are even stronger than in the HDP-HMM of Teh et al. [162]. We instead consider a truncated approximation to the sticky HDP-HMM and develop a sampler that harnesses efficient dynamic programming techniques to block sample the HMM mode sequence. Specifically, we utilize a variant of the *forward-backward algorithm* [139]. Such sampling techniques have been proposed for the finite HMM [148], with analysis showing that blocked sampling requires more computation time but leads to faster mixing rates than a direct sampler.

Learning Multimodal Emissions and Application to Speaker Diarization

Having developed the sticky HDP-HMM framework that accounts for temporal mode persistence, one can examine extending the model to account for multimodal emission distributions. Specifically, we consider Bayesian nonparametric learning of the emission distributions by treating each as a mixture of Gaussians with a Dirichlet process prior. The sticky HDP-HMM's bias towards generating sequences of observations from the same latent HMM mode allows the model to disambiguate the underlying emission distribution. In contrast, a similar extension of the HDP-HMM of Teh et al. [162] to allow multimodal emissions exhibits considerable uncertainty in the choice between rapidly switching amongst HMM modes with single Gaussian emissions or creating persistent HMM modes and associating multiple Gaussian emission components. As a motivating example, we consider the problem of speaker diarization and demonstrate that the sticky HDP-HMM provides state-of-the-art speaker diarizations. We show that such performance relies on the augmented model's ability to capture mode persistence and multimodal emissions.

Chapter Outline

The chapter begins with a review of the HDP-HMM of Teh et al. [162], as well as a demonstration that this model inadequately captures the temporal mode persistence present in many real datasets. We then describe our proposed sticky HDP-HMM framework and how one may place a prior on this self-transition bias parameter and infer it from the data. Both the direct assignment sampler of Teh et al. [162] and the blocked sampler we develop utilizing the truncated sticky HDP-HMM are subsequently outlined. The second half of the chapter focuses on extending the sticky HDP-HMM to allow for Bayesian nonparametric learning of multimodal emission distributions. We conclude with an analysis of the NIST speaker diarization database [126].

■ 1.1.3 Chapter 4: Bayesian Nonparametric Learning of SLDS

Extending the Sticky HDP-HMM to Models with Conditionally Linear Dynamics

The fourth chapter extends the sticky HDP-HMM model of Chapter 3 to scenarios in which a Markov switching model with conditionally linear dynamics provides a better approximation to the observed dynamics than the HMM's assumption of conditionally independent observations. We consider two such models: the switching linear dynamical system (SLDS) and switching vector autoregressive (VAR) process and refer to our Bayesian nonparametric versions of these models as the HDP-SLDS and HDP-AR-HMM, respectively. The basic formulation we present uses a conjugate matrix-normal inverse-Wishart (MNIW) [183] prior on the set of dynamic parameters assuming a fixed model order (i.e., dimension of the SLDS continuous state vector or the autoregressive order.) For the HDP-SLDS and HDP-AR-HMM, we examine a set of synthetic datasets demonstrating our ability to learn switching dynamical models with varying numbers of dynamical regimes. We also examine our ability to segment a sequence of honey bee dances (see Fig. 1.1(b)) and to detect changes in volatility of the IBOVESPA stock index, showing performance competitive with alternative methods and consistent with domain expert analysis.

Sparsity Inducing Priors for Model Order Identification

A more complete system identification of the switching dynamical models that we consider would also involve learning the model order. Although our HDP-SLDS and HDP-AR-HMM formulations assume that, respectively, the underlying state dimension or autoregressive order are fixed, we propose using *automatic relevance determination* (ARD) [9, 112, 124] as a sparsity-inducing prior in place of the conjugate MNIW prior. We specifically encourage mode-specific sparsity in the dynamic parameters in a structured manner that leads to insight into components of the fixed-dimension state vector or fixed set of autoregressive components that do not contribute to the underlying dynamics of the observed phenomenon. In addition to such insights, the sparsity-inducing prior leads to improved parameter estimation in the presence of limited data. We apply this model to a sequence of the honey bee dances, and demonstrate that the turning dances are well-modeled by a single autoregressive component while the waggle dance relies on two components.

Efficient Inference Leveraging Kalman Filtering

Just as we harnessed dynamic programming techniques in the truncated sticky HDP-HMM blocked Gibbs sampler, for the HDP-SLDS we can leverage the conditionally linear dynamics induced by a fixed mode sequence and incorporate efficient Kalman filter computations to block-sample the latent state sequence. Such block sampling of the state sequence was proposed for the finite SLDS in [25]. A later paper [26] analyzed the benefits of an alternative sampler that sequentially samples the dynamical mode sequence, analytically marginalizing the state sequence. We propose a sampler that iterates between block sampling of the mode and state sequences, occasionally interleaving a step of sequentially sampling the mode sequence.

Chapter Outline

Chapter 4 begins with a description of our proposed HDP-SLDS and HDP-AR-HMM dynamical models. We then describe two possible priors for the dynamic parameters: the MNIW prior and the sparsity-inducing ARD prior. We outline our Gibbs sampling algorithm for both the HDP-SLDS and HDP-AR-HMM under these two choices of priors. Simulations on synthetic data and a sequence of honey bee dances demonstrate that the developed HDP-SLDS and HDP-AR-HMM are able to infer both the number of dynamical modes and the underlying model order. We conclude by presenting variants of these models that are commonly found in application areas such as econometrics and target tracking. For the latter application, an alternative sampler harnessing the specific structure of the model is also presented. We present results for the model variants on the IBOVESPA stock index, and synthetic maneuvering target tracking data.

■ 1.1.4 Chapter 5: Sharing Features among Dynamical Systems with Beta Processes

Transferring Knowledge Among Multiple Related Time Series

The final main chapter of the thesis focuses on methods of transferring knowledge between multiple related time series. We assume that each of the time series can be modeled according to the switching dynamical processes of Chapters 3 and 4. We then envision a large library of behaviors, with each time series exhibiting a subset of these behaviors. Specifically, we examine the beta process [67, 165] as a method of tying together the set of behaviors associated with the time series. This process encourages sharing in the chosen behaviors while allowing time-series-specific variability.

One could imagine an alternative architecture based upon the hierarchical Dirichlet process, similar to the model considered in Chapter 4. Specifically, consider a set of HDP-SLDS's tied together by sharing the same set of transition distributions and dynamic parameters. Such a model would assume that each time series was performing exactly the same set of behaviors, and switching between them in the same manner. In addition to allowing each time series to choose a unique subset of the full set of behaviors, our proposed model using the beta process prior also enables multiple time series to select the same set of behaviors, but to switch between them in a unique manner. To test our proposed model, we analyze a set of exercise routine videos from the Carnegie Melon University (CMU) motion capture database [169] and demonstrate that we are indeed able to identify common motion behaviors. A benefit of our Bayesian nonparametric approach is that we are also able to discover motions unique to a given video.

Birth-Death RJ-MCMC for Non-Conjugate IBP Models

The model we introduce does not allow for conjugate analysis, and previous samplers for the non-conjugate case either relied on approximations [59] or proposals from the prior [117] that result in low acceptance rates in high-dimensional applications. In contrast, we develop a Markov chain Monte Carlo (MCMC) sampler that uses reversible jump [60] birth and death proposals to explore the incorporation of new behaviors, and exploits the finite dynamical system induced by a fixed set of behaviors to efficiently compute acceptance probabilities.

Chapter Outline

We start by describing how the beta process may be used as a prior for relating the switching dynamical processes of Chapters 3 and 4. Having established the generative model, we describe an MCMC inference algorithm that allows for efficient exploration of the set of possible behaviors. We conclude the chapter with empirical results on a set of synthetic data, and on data from the CMU motion capture database.

■ 1.1.5 Chapter 6: Contributions and Recommendations

We conclude by surveying the contributions of this thesis, and highlights of directions for future research. Each chapter concludes with a lengthy discussion of areas of future research. In this chapter we simply abstract and jointly examine common themes appearing throughout the thesis.

■ 1.1.6 Appendices

For readability and clarity of the main concepts of the thesis, the majority of derivations are placed in a series of appendices appearing at the end of the thesis. These derivations focus on determining the conditional distributions and message passing schemes used in our MCMC samplers, and rely heavily upon the background material presented in Chapter 2.

Chapter 2

Background

IN this background chapter, we review the statistical methodologies upon which our contributions are based. We begin in Sec. 2.1 by motivating the Bayesian framework through a discussion of exchangeability and *de Finetti's theorem*, which can be viewed as a justification for the use of prior distributions. We then describe exponential families of probability distributions and sufficient statistics in Sec. 2.2 and Sec. 2.3, respectively. Together, these concepts enable examination of prior distributions, namely conjugate prior distributions, that lead to efficient inference techniques, as discussed in Sec. 2.4.

In Sec. 2.5, we turn to discussing the graphical model representation of joint probability distributions that allows for the development of efficient inference techniques. We first provide an introduction to graphical models, with an emphasis on the directed chains and their associated inference techniques that provide the basis for the models we consider and are fundamental to our derivations. In Sec. 2.6 and Sec. 2.7, we specifically consider two such simple directed chain graphical models that are the basic building blocks for the more complex models we consider in this thesis: the hidden Markov model and the state space model. For each of these models, we provide an interpretation of their associated classical inference techniques in terms of the general graphical model framework described in Sec. 2.5.

For the models we consider in this thesis, exact inference is infeasible and we rely on Markov chain Monte Carlo techniques that are outlined in Sec. 2.8. Finally, we conclude in Sec. 2.9 by providing background material on the stochastic processes we use in developing our Bayesian nonparametric models: the Dirichlet process, its hierarchical extension, and the beta process.

■ 2.1 The Bayesian Framework

In this section we provide a brief motivation for the Bayesian approach and establish some concepts that reappear throughout this thesis. The overarching goal of the thesis is then to examine the flexibility a Bayesian approach can provide in the case of learning dynamical systems.

2.1.1 Modeling via Exchangeability

The concept of exchangeability is central to many statistical approaches, and may be viewed as critical in motivating Bayesian statistics. Let us assume that we are aggregating data in an attempt to make predictions about future values of the random process we are observing. If we were to make the strong assumption of the data being *independent*, we would treat every new data point individually without using past observations to predict future observations since:

$$p(y_1, \ldots, y_n) = \prod_{i=1}^{n} p(y_i) \tag{2.1}$$

implies that

$$p(y_{n+1}, \ldots, y_m \mid y_1, \ldots, y_n) = p(y_{n+1}, \ldots, y_m). \tag{2.2}$$

A weaker assumption that often better describes the data we encounter is that of *exchangeability*, which states that the order we encounter the data is inconsequential.

Definition 2.1.1. *A sequence of random variables $y_1, y_2, \ldots, y_n$ is said to be* finitely exchangeable *if*

$$y_1, y_2, \ldots, y_n \overset{\mathcal{D}}{=} y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(n)} \tag{2.3}$$

for every permutation π on $\{1, \ldots, n\}$. Here, we use the notation $\overset{\mathcal{D}}{=}$ to mean equality in distribution.

From this definition, we see that independence implies exchangeability, but not vice versa. We are often in settings where data is continually accumulated, or in which fixing an upper bound n is challenging. We would thus like to formalize a notion of exchangeability for infinite sequences.

Definition 2.1.2. *A sequence $y_1, y_2, \ldots$ is said to* infinitely exchangeable *if every finite subsequence is finite exchangeable [15].*

As is demonstrated in Bernardo and Smith [15], not every finitely exchangeable sequence can be embedded in an infinitely exchangeable sequence.

Example 2.1.1. *As an example of infinite exchangeability, consider an urn with b black balls and w white balls. Draw a ball at random from the urn and replace that ball along with n balls of the same color. Continue repeating this procedure infinitely many times. Such an urn is typically referred to as a* Pólya urn. *Let $y_i = 1$ if the i^{th} draw from the urn produces a black ball, and $y_i = 0$ otherwise. Then,*

$$\begin{aligned} p(1,1,0,1) &= \frac{b}{b+w}\frac{b+n}{b+w+n}\frac{w}{b+w+2n}\frac{b+2n}{b+w+3n} \\ &= \frac{b}{b+w}\frac{w}{b+w+n}\frac{b+n}{b+w+2n}\frac{b+2n}{b+w+3n} \\ &= p(1,0,1,1). \end{aligned}$$

The denominator is the same for all possible sequences since n balls are added at every draw regardless of the color of the drawn ball. The sequence of terms in the numerator simply depends upon how many previous times a black or white ball was drawn, not the specific order. Using this argument, one can prove that every finite subsequence of data generated from this urn procedure are exchangeable under this model. However, we can clearly see that the data are not independent, nor even a Markov process.

Exchangeability has simplifying implications for inference since we can simply ignore the order in which the data arrive. Sometimes, however, exchangeability is too strong of an assumption. Relaxations include considering *partially exchangeable* data where some auxiliary information partitions the data into exchangeable sets. For example, consider a person flipping two biased coins, one on even throws and the other on odd throws. The data are exchangeable within the set of odd or even tosses if these labels are provided. There are many possible extensions and variations on the standard exchangeability model; however, the end goal is to group data into exchangeable, and thus relatively simple, blocks for which inference is more tractable.

A very important result arising from the assumption of exchangeable data is what is typically referred to as *de Finetti's theorem.* This theorem states that an infinite sequence of random variables $y_1, y_2, \dots$ is exchangeable if and only if there exists a random probability measure ν with respect to which $y_1, y_2, \dots$ are conditionally i.i.d. with distribution ν. Furthermore, this random measure can be viewed as the limiting empirical measure. De Finetti actually proved this in the case of binary random variables de Finetti [33], with the more general extension to arbitrary real-valued exchangeable sequences made by Hewitt and Savage [66] and Ryll-Nardzewski [146].

Theorem 2.1.1. *If $y_1, y_2, \dots$ is an infinitely exchangeable sequence of binary random variables with probability measure P, then there exists a distribution function Q on* $[0, 1]$ *such that for all n*

$$p(y_1, \dots, y_n) = \int_0^1 \prod_{i=1}^n \vartheta^{y_i}(1-\vartheta)^{1-y_i} dQ(\vartheta), \tag{2.4}$$

where $p(y_1, \dots, y_n)$ is the joint probability mass function defined by measure P. Furthermore, Q is the distribution function of the limiting empirical frequency:[1]

$$\theta \overset{a.s.}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^n y_i, \quad \theta \sim Q. \tag{2.5}$$

Proof. Originally presented in [33]. See Bernardo and Smith [15] and Heath and Sudderth [65] for a proof in more modern terms. ■

[1] The notation $x \sim F$ indicates that the random variable x is drawn from a distribution F. We use bar notation $x \mid F \sim F$ to specify conditioned upon random variables, such as a random distribution.

This theorem can be interpreted as saying that if $y_1, y_2, \ldots$ is an infinitely exchangeable binary sequence, then it is as if the elements of this sequence are independent Bernoulli random variables with probability of success θ, where θ has distribution Q. Furthermore, one can interpret Q as our belief about the limiting empirical frequency of ones in the data.

From de Finetti's theorem, we see the motivation for the Bayesian perspective of the parameter yielding the observations i.i.d. as a random quantity with some distribution Q, rather than as a fixed and unknown quantity. We now state the more general form of the de Finetti theorem.

Theorem 2.1.2. *If $y_1, y_2, \ldots$ is an infinitely exchangeable sequence of real-valued random variables with probability measure P, then there exists a probability measure μ defined on the space of all probability measures $\mathcal{P}(\mathbb{R})$ on $\mathbb{R}$ such that*[2]

$$P(y_1 \in A_1, \ldots, y_n \in A_n) = \int_{\mathcal{P}(\mathbb{R})} \prod_{i=1}^{n} \mathcal{V}(A_i)\mu(d\mathcal{V}) \tag{2.6}$$

Furthermore, μ is the law of a probability measure ν, where ν is almost surely defined by the limiting empirical measure. Namely,

$$\nu(B) \overset{a.s.}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_B(y_i), \quad \nu \sim \mu. \tag{2.7}$$

where B ranges over all elements of the Borel σ-algebra. The measure μ is often referred to as the de Finetti measure.

Proof. See Hewitt and Savage [66] and Ryll-Nardzewski [146]. ■

From a generative perspective, the theorem states that if $y_1, y_2, \ldots$ are infinitely exchangeable, then there exists a measure μ on measures such that:

$$\begin{aligned} \nu &\sim \mu \\ y_i \mid \nu &\overset{i.i.d.}{\sim} \nu. \end{aligned} \tag{2.8}$$

When we take the sets A_i to be $(-\infty, y_i]$, we obtain a form of the above theorem in terms of the random distribution functions F associated with the random measures ν.

Example 2.1.2. *As an informal presentation to provide some intuition for this theorem, let us return to the case of binary random variables. Assume the phenomenon we are observing are realizations from a game, though we do not know the game being played. Instead, we are simply observers of the outcomes of the game. For example,*

[2] Here, we use $\mathcal{V}$ as the variable of integration when integrating with respect to the probability measure μ. We then use ν as the random measure with law μ.

assume the observed phenomenon are flips of a coin with probability p of heads. The limiting empirical frequency of heads (i.e., 1's) in infinitely many draws will be p almost surely. The de Finetti theorem then implies that the de Finetti measure μ is degenerate on

$$\nu = p\delta_1 + (1-p)\delta_0$$

because every such infinite sequence of flips of that coin results in the same empirical measure. See Fig. 2.1(a). Here, we use δ_i to be a measure concentrated at i.

On the other hand, assume we are observing draws from a Pólya urn starting with b black balls and w white balls, adding n balls per round, as in Example 2.1.1. From this example, we know that the data are exchangeable. We observe an infinite binary sequence which gives us the following empirical measure:

$$\nu_1 = \theta_1\delta_1 + (1-\theta_1)\delta_0.$$

We are provided with infinitely many such sequences from an urn in the (b, w) starting configuration (i.e., infinitely many realizations from this game.) For each infinite sequence, we build the empirical measure

$$\nu_i = \theta_i\delta_1 + (1-\theta_i)\delta_0 \quad i = 1, 2, \ldots.$$

De Finetti tells us that these ν_i are instantiations of the random measure ν. In essence, we can empirically build up the de Finetti measure μ by examining the infinite collection of empirical measures. Let us instead examine the distribution *Q on θ. One can show that Q is a Beta$(b/n, w/n)$ distribution (see Sec. 2.4.2), as empirically demonstrated in Fig. 2.1(b), implying that the generative process is*

$$\theta \sim Beta(b/n, w/n)$$
$$y_i \mid \theta \overset{i.i.d.}{\sim} Ber(\theta),$$

where Ber denotes the Bernoulli distribution. This process is referred to as a Beta-Bernoulli *process. There are many other such games for generating infinitely exchangeable binary sequences that we could be observing, each corresponding to a different de Finetti measure. As the observer of these sequences, de Finetti simply tells us that there* exists *a random probability measure which yields the data i.i.d.; we would need to observe infinitely many sequences to actually* reconstruct *the distribution on this probability measure associated with the underlying game.*

We have seen in Theorem 2.1.1 that for infinitely exchangeable binary sequences, there exists a random probability measure ν that concentrates on $\{0, 1\}$ implying that this measure can be uniquely described by a single parameter θ. One can straightforwardly extend the argument in Theorem 2.1.1 to infinitely exchangeable sequences taking values in $\{1, \ldots, K\}$; here, the random measure yielding the data i.i.d. concentrates on $\{1, \ldots, K\}$ and is thus uniquely defined by a $(K-1)$-dimensional parameter

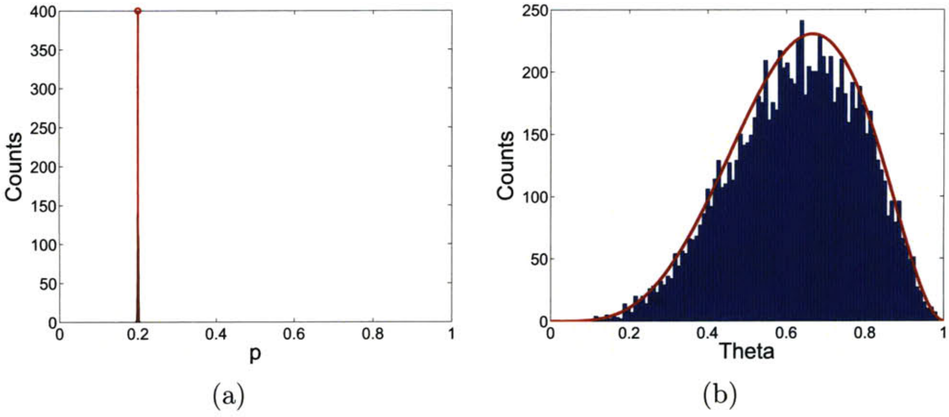


Figure 2.1. (a) Histogram of the empirical estimates of the probability of heads, p, in the coin-flipping experiment from 10,000 trials. Each trial's estimate is based on 100,000 observations. The red line indicates the true probability of heads. (b) Histogram of the empirical estimates of the parameter θ that yields the exchangeable observations drawn from a Pólya urn i.i.d.. The histogram is the result of 10,000 trials from an urn starting with 10 black balls and 6 white balls, and replacing 2 balls at every draw from the urn. Each trial's estimate of θ is produced based on 1,000 observations. The red line indicates a Beta$(10/2, 6/2)$ distribution.

$\theta = \{\theta_1, \dots, \theta_{K-1}\}$ [15]. Analogous to the examples presented in Example 2.1.2, possible underlying games include rolling a K-sided weighted die or drawing from an urn with K different colored balls. When moving to infinitely exchangeable sequences taking values in the reals, the random probability measures ν can be arbitrarily complex and are, in general, defined by infinitely many parameters (i.e., ν is a generic element of $\mathcal{P}(\mathbb{R})$.) Some special cases exist in which the parametrization remains finite. For example, if ν is almost surely a Gaussian distribution, the parametrization solely consists of a mean and variance. The more general case in which θ may be an infinite-dimensional parameter motivates the development of Bayesian nonparametric methods, some of which we explore in this thesis. For example, the Dirichlet process of Sec 2.9.1 defines a distribution on probability measures that concentrate at a countably infinite number of elements of the reals (or the more general spaces we consider in Sec. 2.9.1.)

When we limit ourselves to the more restrictive class of finite-dimensional θ (e.g., Bernoulli, multinomial, Gaussian random variables), we can invoke the following corollaries.

Corollary 2.1.1. *Assuming the required densities exist, and assuming the conditions of Theorem 2.1.2 hold, then there exists a distribution function Q such that the joint density of $y_1, \dots, y_n$ is of the form*

$$p(y_1, \dots, y_n) = \int_\Theta \prod_{i=1}^{n} p(y_i \mid \vartheta) dQ(\vartheta), \tag{2.9}$$

with $p(\cdot \mid \vartheta)$ representing the density function corresponding to the finite-dimensional *parameter* $\vartheta \in \Theta$.

From the above corollary, it is simple to see how the de Finetti theorem motivates the concept of a *prior distribution* $Q(\cdot)$ and a *likelihood function* $p(y \mid \cdot)$.

Corollary 2.1.2. *Given that the conditions of Corollary 2.1.1 hold, then the* predictive density *is given by*

$$p(y_{m+1}, \ldots, y_n \mid y_1, \ldots, y_m) = \int_\Theta \prod_{i=m+1}^{n} p(y_i \mid \vartheta) dQ(\vartheta \mid y_1, \ldots, y_m), \tag{2.10}$$

where

$$dQ(\theta \mid y_1, \ldots, y_m) = \frac{\prod_{i=1}^{m} p(y_i \mid \theta) dQ(\theta)}{\int_\Theta \prod_{i=1}^{m} p(y_i \mid \vartheta) dQ(\vartheta)}. \tag{2.11}$$

Proof. The result follows directly from employing

$$p(y_{m+1}, \ldots, y_n \mid y_1, \ldots, y_m) = \frac{p(y_1, \ldots, y_n)}{p(y_1, \ldots, y_m)},$$

along with Corollary 2.1.1. ■

From the form of the predictive density in Eq. (2.10), we see that our view of the existence of an underlying random parameter θ yielding the data i.i.d. has not changed. Instead, we have simply updated our *prior belief* $Q(\theta)$ into a *posterior belief* $Q(\theta \mid y_1, \ldots, y_m)$ through an application of *Bayes rule*:

$$p(\theta \mid y) = \frac{p(y \mid \theta) p(\theta)}{\int_\Theta p(y \mid \vartheta) p(\vartheta) d\vartheta} = \frac{p(y \mid \theta) p(\theta)}{p(y)}. \tag{2.12}$$

Here, we have written the rule in its simplest form assuming that a density on θ exists in addition to the conditional density on y. Although one can view the computation of the predictive distribution in Eq. (2.10) as the objective in Bayesian statistics, we will often limit our discussion to the process of forming the posterior distribution in Eq. (2.11) from the prior by incorporating observations, since this is a fundamental step in examining the predictive distribution.

From a practical perspective, we never have an infinite sequence of observations from which to characterize our prior distribution. Furthermore, even if we had such a quantity, the probability measure that the de Finetti theorem would suggest as yielding the data i.i.d. might be arbitrarily complex. Thus, we are left with two competing pragmatic choices in defining our prior:

(1) Tractable inference,

(2) Modeling flexibility.

The issue of tractable inference often motivates the use of conjugate priors, as discussed in Sec. 2.4. The goal of flexibility in our models leads to the study of Bayesian nonparametric methods. A brief introduction to some specific classes of nonparametric methods that maintain computational tractability is presented in Sec. 2.9.1-Sec. 2.9.4.

Another key aspect of the Bayesian framework we have established is in characterizing a model, or *likelihood distribution*, $p(y \mid \theta)$ for how our data are generated conditioned a parameter value θ. This choice, too, is often motivated by practical considerations that are typically coupled with those of choosing a prior distribution. We do not develop a full analysis of model selection in this thesis, but begin the exploration in Sec. 2.2. As practitioners, we do not actually know the underlying generative process, but we can use a combination of our insight on the process (e.g., we know we are observing heights from a given population and heights tend to be well-modeled as Gaussian) and our adherence to computational limitations to define a model.

■ 2.2 Exponential Families

Exponential families represent a fundamental class of distributions in statistics. They arise as the answer to numerous, albeit related, questions. Within the Bayesian framework: For what class of models does there exist a prior that leads to computationally tractable inference [15, 141]? Frequentists arrive at the exponential family when asking: If there exists an efficient estimator, can we describe the class of models from which the data could have been generated [87, 184]? Common to both domains: What distribution is maximally random while being consistent with a set of moment constraints [15, 79, 116]?

Definition 2.2.1. *A parametrized family of distributions $\mathcal{P}_\Theta = \{P_\theta\}$ is a* k-parameter exponential family *with natural parameter $\boldsymbol{\eta}(\cdot) = [\eta_1(\cdot), \dots, \eta_k(\cdot)]^T$, natural statistic $\boldsymbol{t}(\cdot) = [t_1(\cdot), \dots, t_k(\cdot)]^T$, and base distribution $q(\cdot) \propto e^{\beta(\cdot)}$ if each member P_θ of the family has a density of the form*

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \exp\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\boldsymbol{t}(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\} \tag{2.13}$$

$$= \exp\left\{\sum_{i=1}^{k} \eta_i(\boldsymbol{\theta}) t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} \tag{2.14}$$

with respect to a dominating measure[3] *μ. Here, $\boldsymbol{y}$*[4] *is a point in the sample space $\mathcal{Y}$, which represents the support of the density. The function $\alpha(\cdot)$ is referred to as the log-partition function and ensures that the probability density integrates to 1. We will denote this family by $\mathcal{E}(\boldsymbol{\theta}; \boldsymbol{\eta}(\cdot), \boldsymbol{t}(\cdot), \beta(\cdot))$.*

[3] The dominating measure is the assumed measure on the considered measurable space, and as such provides the measure with respect to which the Radon-Nikodym derivative is taken when defining densities (amongst other measure-theoretic operations one could examine).

[4] We use the notation $\boldsymbol{y}$ rather than y to indicate that this quantity is allowed to be vector valued.

The set of admissible parameter values, or the *natural parameter space*, for which a constant $\alpha(\boldsymbol{\theta})$ exists are those such that

$$\int \exp\left\{\sum_{i=1}^{k} \eta_i(\boldsymbol{\theta})t_i(\boldsymbol{y}) + \beta(\boldsymbol{y})\right\} dy < \infty. \tag{2.15}$$

We could generalize Eq. (2.15) and the results to follow for a given measure μ rather than the assumed Lebesgue (or where appropriate, counting) measure. However, we will omit this level of mathematical formality.

It is common to restrict oneself to examining families of distributions whose support, i.e., the set of $\boldsymbol{y}$ such that $p(\boldsymbol{y} \mid \boldsymbol{\theta}) > 0$, does not depend upon $\boldsymbol{\theta}$.

Definition 2.2.2. *An exponential family $\mathcal{E}(\boldsymbol{\theta}; \boldsymbol{\eta}(\cdot), \boldsymbol{t}(\cdot), \beta(\cdot))$ is called* regular *if the support of each member of the family does not depend upon the value of the parameter $\boldsymbol{\theta}$.*

Another form of exponential families that deserves a special name is when the density of each member of the family depends linearly on the parameters, i.e., $\boldsymbol{\eta}(\boldsymbol{\theta}) = [\theta_1, \dots, \theta_k]$.

Definition 2.2.3. *A* canonical *exponential family is one which depends linearly on the parameter $\boldsymbol{\theta}$:*

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\theta}) &= \exp\{\boldsymbol{\theta}^T \boldsymbol{t} - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\} && (2.16)\\ &= \exp\left\{\sum_{i=1}^{k} \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\}. && (2.17) \end{aligned}$$

We will denote the canonical exponential family by $\mathcal{E}(\boldsymbol{\theta}; \boldsymbol{I}(\cdot), \boldsymbol{t}(\cdot), \beta(\cdot))$.

One, in theory, can always consider an exponential family in its canonical form by defining a family $\mathcal{P}_\eta$ with the parameters as the possibly nonlinear mapping $\boldsymbol{\eta}(\boldsymbol{\theta}) \triangleq [\eta_1, \dots, \eta_k]$ and the log-partition function $\alpha(\cdot)$ appropriately redefined. In practice, however, it might be challenging to find the set of admissible values of $\boldsymbol{\eta}$ and the form of the log-partition function. Note that some references, such as Bernardo and Smith [15], use the term *canonical* to refer to exponential families that also depend linearly on the data.

Definition 2.2.4. *For data $\boldsymbol{y}$ distributed according to $p(\boldsymbol{y} \mid \boldsymbol{\theta})$, a parameter $\boldsymbol{\theta}$ is termed* unidentifiable *on the basis of $\boldsymbol{y}$ if there exists $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ such that $P_{\theta_1} = P_{\theta_2}$.*

Lemma 2.2.1. *If the set of natural statistics $[t_1(\cdot), \dots, t_k(\cdot)]$ are linearly dependent, then the parameters $[\eta_1, \dots, \eta_k]$ are unidentifiable from the data $\boldsymbol{y}$.*

Proof. Assume, without loss of generality, that $t_k(\boldsymbol{y}) = ct_{k-1}(\boldsymbol{y})$ for some constant c. Take $\eta'_i = \eta_i$ for $i = 1, \ldots, k-2$, $\eta'_{k-1} = \eta_{k-1} + c\eta_k$, and $\eta'_k = 0$. Then,

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\eta}) &= \exp\left\{\sum_{i=1}^{k} \eta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} \\ &= \exp\left\{\sum_{i=1}^{k-2} \eta_i t_i(\boldsymbol{y}) + (\eta_{k-1} + c\eta_k)t_{k-1}(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} \\ &= \exp\left\{\sum_{i=1}^{k-2} \eta'_i t_i(\boldsymbol{y}) + \eta'_{k-1} t_{k-1}(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} \\ &= p(\boldsymbol{y} \mid \boldsymbol{\eta}'). \end{aligned}$$

■

From the above, we see that whenever there are linearly dependent natural statistics, we can find an equivalent, reduced-order exponential family. We will assume that we always restrict ourselves to such reduced-order models. Note that the same issue arises if the components of the natural parameter $\boldsymbol{\eta}(\boldsymbol{\theta})$ are linearly dependent functions of $\boldsymbol{\theta}$.

Definition 2.2.5. *A* minimal *exponential family is one in which there does not exist a non-zero vector $\boldsymbol{a} = [a_1, \ldots, a_k]$ such that*

$$\sum_{i=1}^{k} a_i t_i(\boldsymbol{y}) \tag{2.18}$$

is equal to a constant.

■ 2.2.1 Properties of the Canonical Exponential Family

The following theorem leads to a number of useful properties of the exponential family, specifically, the moment-generating property of the log-partition function.

Theorem 2.2.1. *For any integrable function $f(\cdot)$ and any $\boldsymbol{\theta}$ in the set of natural parameters, the integral*

$$\int f(\boldsymbol{y}) \exp\left\{\sum \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} d\boldsymbol{y} \tag{2.19}$$

is continuous and has derivatives of all orders with respect to the parameters $\boldsymbol{\theta}$.

Proof. See Barndorff-Nielsen [8], amongst other texts. ■

Corollary 2.2.1. *The expected value and covariance of the natural statistics $t_i(\boldsymbol{y})$ are related to derivatives of the log-partition function by*

$$\mathbb{E}_{\boldsymbol{\theta}}[t_i(\boldsymbol{y})] = \frac{\partial}{\partial \theta_i}\alpha(\boldsymbol{\theta}) \tag{2.20}$$

and

$$cov(t_i(\boldsymbol{y}), t_j(\boldsymbol{y})) = \frac{\partial^2}{\partial \theta_j \partial \theta_i}\alpha(\boldsymbol{\theta}), \tag{2.21}$$

respectively.

Proof. We apply Theorem 2.2.1 to the following identity, arising from the unit integrability of the density $p(\cdot \mid \boldsymbol{\theta})$:

$$\int \exp\left\{\sum \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} d\boldsymbol{y} = 1.$$

Taking the derivative with respect to θ_i,

$$\int t_i(\boldsymbol{y}) \exp\left\{\sum \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} d\boldsymbol{y} = \frac{\partial}{\partial \theta_i}\alpha(\boldsymbol{\theta}). \tag{2.22}$$

The first equality of the corollary results from noting that the left-hand side of Eq. (2.22) is the expected value of $t_i(\boldsymbol{y})$ under the given exponential family. Differentiating again with respect to θ_j yields

$$\begin{aligned}\int \left(t_i(\boldsymbol{y}) - \frac{\partial}{\partial \theta_i}\alpha(\boldsymbol{\theta})\right)\left(t_j(\boldsymbol{y}) - \frac{\partial}{\partial \theta_j}\alpha(\boldsymbol{\theta})\right) \exp\left\{\sum \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} d\boldsymbol{y} \\ - \frac{\partial^2}{\partial \theta_j \partial \theta_i}\alpha(\boldsymbol{\theta}) \int \exp\left\{\sum \theta_i t_i(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\right\} d\boldsymbol{y} = 0.\end{aligned} \tag{2.23}$$

Identifying the first line of Eq. (2.23) as $\text{cov}(t_i(\boldsymbol{y}), t_j(\boldsymbol{y}))$ using the fact that $\frac{\partial}{\partial \theta_j}\alpha(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[t_j(\boldsymbol{y})]$, and identifying the integral of the second line as 1 completes the proof. ■

The above implies that instead of computing potentially complicated integrals, we can find the moments of a distribution by calculating the derivatives of the log-partition function. We note, however, that finding the log-partition function is often a challenge in and of itself. Another important implication of the above result is that since $\nabla^2\alpha(\cdot)$ is a positive semi-definite covariance matrix, $\alpha(\boldsymbol{\theta})$ is a convex function in θ. For minimal exponential families, $\nabla^2\alpha(\cdot)$ must be positive definite, implying strict convexity. Such interpretations of $\alpha(\cdot)$ have important implications for the geometry of exponential families that are exploited in variational approaches [176].

■ 2.2.2 Interpretation as Linearly Constrained Maximum Entropy Distribution

As alluded to at the beginning of this section, the exponential family can be derived as the maximally random distribution subject to a set of linear constraints. To derive this result, and to formalize our definition of *randomness*, we need to rely on some information-theoretic concepts. See Cover and Thomas [31] for a more detailed exploration of these terms.

Fundamental Quantities of Information Theory

Shannon's measure of *entropy* conveys the uncertainty of a discrete random variable y taking values within a finite space $\mathcal{Y}$:

$$H(y) = -\sum_{y \in \mathcal{Y}} p(y) \log p(y), \tag{2.24}$$

where $p(y)$ is the associated probability mass function defining the law of y. If the log is base 2, the units of this measure is in bits while for base e the units are nats. From this definition, one can easily prove that

$$0 \leq H(y) \leq \log |\mathcal{Y}|. \tag{2.25}$$

One can extend the idea of entropy to jointly random variables $(x, y) \sim p(x, y)$, $x \in \mathcal{X}$, in which case the *joint entropy* is defined as

$$H(x, y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y), \tag{2.26}$$

One can similarly define the *conditional entropy* of a random variable y given x:

$$H(y \mid x) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y \mid x). \tag{2.27}$$

Using standard manipulations, one can show that the joint entropy $H(x, y)$ is simply the sum of the entropy of x, $H(x)$, and the conditional entropy of y given x, $H(y \mid x)$, which has a nice interpretation in terms of conservation of uncertainty. The change in entropy of a random variable y after an observation x is given by the *mutual information*

$$\begin{aligned} I(y; x) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} && (2.28) \\ &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y)(\log p(y) - \log p(y \mid x)) && (2.29) \\ &= H(y) - H(y \mid x). && (2.30) \end{aligned}$$

The above definitions can be extended to continuous random variables by considering *differential entropy*

$$h(\boldsymbol{y}) = -\int_{\mathcal{Y}} p(\boldsymbol{y}) \log p(\boldsymbol{y}) d\boldsymbol{y}, \tag{2.31}$$

and *differential conditional entropy*

$$h(\boldsymbol{y} \mid \boldsymbol{x}) = -\int_{\mathcal{X}} \int_{\mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \log p(\boldsymbol{y} \mid \boldsymbol{x}) d\boldsymbol{y} d\boldsymbol{x}. \tag{2.32}$$

However, although discrete entropy is a non-negative quantity, differential entropy does not have this property.

Finally, we define a measure of the distance between two densities p and q. The *relative entropy* or *Kullback-Leibler (KL) divergence* is given by:

$$D(p||q) = \int_{\mathcal{Y}} p(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{q(\boldsymbol{y})} d\boldsymbol{y}. \tag{2.33}$$

Note that because KL divergence is not symmetric, it is not actually a distance metric. From this definition, we see that mutual information can be interpreted as the KL divergence between a joint distribution of $(\boldsymbol{x}, \boldsymbol{y})$ and the distribution assuming they are independent random variables:

$$I(\boldsymbol{y}; \boldsymbol{x}) = D(p(\boldsymbol{x}, \boldsymbol{y}) || p(\boldsymbol{x}) p(\boldsymbol{y})). \tag{2.34}$$

Here, the mutual information is defined in terms of differential entropy.

Projections onto Exponential Families

Let us define a *linear family* of distributions for a random variable $\boldsymbol{y}$ as

$$\mathcal{L}_t = \{p : \mathbb{E}_p[t_k(\boldsymbol{y})] = \mu_k, \quad k = 1, \ldots, K\}. \tag{2.35}$$

This family is termed linear since for all $p_1, p_2 \in \mathcal{L}_t$, and for all $\lambda \in [0,1]$, $p_\lambda = \lambda p_1 + (1-\lambda) p_2 \in \mathcal{L}_t$.

Theorem 2.2.2. *Let $\boldsymbol{t}(\cdot) = \{t_1(\cdot), \ldots, t_K(\cdot)\}$ be a set of functions defined on $\mathcal{Y}$ and $\{\mu_1, \mu_2, \ldots, \mu_K\}$ be a set of arbitrary constants. Define the linear family*

$$\mathcal{L}_t = \{p : \mathbb{E}_p[t_k(\boldsymbol{y})] = \mu_k, \quad k = 1, \ldots, K\}, \tag{2.36}$$

and consider the element of this family, p^*, which satisfies

$$p^* = \arg \min_{p \in \mathcal{L}_t} D(p||q), \tag{2.37}$$

where the support of q contains that of p. Then p^* belongs to the exponential family

$$\mathcal{E}_t = \left\{ p : p(\boldsymbol{y}) = \exp\left(\sum_{k=1}^{K} \lambda_k t_k(\boldsymbol{y}) - \alpha(\boldsymbol{\lambda}) + \log q(\boldsymbol{y}) \right) \right\}. \tag{2.38}$$

Proof. See Bernardo and Smith [15] for a proof. The basic idea is to minimize the Lagrange function consisting of the KL divergence and a set of Lagrange multipliers that enforce the linear constraints, as well as the constraint that p^* must be a valid density. These Lagrange multipliers end up as the natural parameters $\boldsymbol{\lambda}$ of the exponential family. ■

If we take $q(\boldsymbol{y}) \propto 1$ for all $\boldsymbol{y} \in \mathcal{Y}$ (an improper distribution in the case when $\mathcal{Y}$ is not finite), then p^* has the interpretation as the *maximum entropy* distribution that satisfies a set of moment constraints. As an example, the maximum entropy distribution over the real line subject to a second moment constraint is a zero-mean Gaussian distribution with variance given by that constraint.

■ 2.2.3 Examples

Many well-known classes of distributions can be cast within the framework of an exponential family. We now present a set of examples of such manipulations.

Bernoulli

$$p(y \mid \theta) = \theta^y (1-\theta)^{1-y} \quad \frac{\theta^y e^{-\theta}}{y!} \quad y \in \{0,1\} \tag{2.39}$$

$$\ln p(y \mid \theta) = y \ln \theta + (1-y)\ln(1-\theta) \tag{2.40}$$

$$= \underbrace{\ln\left(\frac{\theta}{1-\theta}\right)}_{\eta(\theta)} \underbrace{y}_{t(y)} + \underbrace{\ln(1-\theta)}_{\alpha(\theta)} \tag{2.41}$$

Geometric

$$p(y \mid \theta) = (1-\theta)\theta^y \quad y \in \{0,1,2,\dots\} \tag{2.42}$$

$$\ln p(y \mid \theta) = \underbrace{\ln(\theta)}_{\eta(\theta)} \underbrace{y}_{t(y)} + \underbrace{\ln(1-\theta)}_{\alpha(\theta)} \tag{2.43}$$

Poisson

$$p(y \mid \theta) = \frac{\theta^y e^{-\theta}}{y!} \quad y \in \{0,1,2,\dots\} \tag{2.44}$$

$$\ln p(y \mid \theta) = \underbrace{\ln(\theta)}_{\eta(\theta)} \underbrace{y}_{t(y)} - \underbrace{\theta}_{\alpha(\theta)} - \underbrace{\ln y}_{\beta(y)} \tag{2.45}$$

Exponential

$$p(y \mid \theta) = \theta e^{-\theta y} \quad y > 0 \tag{2.46}$$

$$\ln p(y \mid \theta) = \underbrace{-\theta y}_{\eta(\theta)t(y)} + \underbrace{\ln \theta}_{\alpha(\theta)} \tag{2.47}$$

■ 2.3 Sufficient Statistics

For the exponential family, we have seen that the densities only depend on the data through the natural statistics $\boldsymbol{t}(\boldsymbol{y})$ and the base distributions $q(\boldsymbol{y}) \propto \exp\{\beta(\boldsymbol{y})\}$. This leads one to ask under what conditions are inferences using transformations of the data, or *statistics*, the same as if we had used the data itself. One might additionally ask what set of models yield a compact set of statistics, summarizing an arbitrarily large set of data, that are *sufficient* for the inferences we wish to make. In the following, we establish a formal framework for this data-processing concept.

Definition 2.3.1. *Given a sequence of random variables $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$, with $\boldsymbol{y}_j \in \mathcal{Y}_j$ and probability measure P, a sequence of statistics $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$, with each function $\boldsymbol{t}_j$ defined on the product space $\mathcal{Y}_1 \times \cdots \times \mathcal{Y}_j$, is said to be* predictive sufficient *for $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ if*

$$p(\boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_k} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_j) = p(\boldsymbol{y}_{i_1}, \ldots, \boldsymbol{y}_{i_k} \mid \boldsymbol{t}_j) \qquad \forall j, k \tag{2.48}$$

where $\{i_1, \ldots, i_k\}$ are a set of indices not seen in $\{1, \ldots, j\}$. Here, $p(\cdot \mid \cdot)$ is the conditional density induced by the measure P.

That is, given $\boldsymbol{t}_j = \boldsymbol{t}_j(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_j)$, the values of the data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_j$ do not further contribute to the prediction of future values of data.

Definition 2.3.2. *Given an exchangeable*[5] *sequence of random variables $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$, each with sample space $\mathcal{Y}$, the sequence of statistics $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$, with each function $\boldsymbol{t}_j$ defined on the product space $\mathcal{Y}^j$, is said to be* parametric sufficient *for $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ if*

$$dQ(\boldsymbol{\theta} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = dQ(\boldsymbol{\theta} \mid \boldsymbol{t}_n) \quad \forall n \geq 1, \tag{2.49}$$

for any $dQ(\boldsymbol{\theta})$ such that

$$p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \int \prod_{i=1}^{n} p(\boldsymbol{y}_i \mid \boldsymbol{\theta}) dQ(\boldsymbol{\theta}). \tag{2.50}$$

Informally, this definition of sufficiency implies that, given exchangeable data, posterior inference using a parametric sufficient statistic results in the same analysis as using the data itself. The following theorem provides a connection between a statistic being sufficient for prediction and for posterior inference.

Theorem 2.3.1. *Given an exchangeable sequence of random variables $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$, each with sample space $\mathcal{Y}$, the sequence of statistics $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$, with $\boldsymbol{t}_j$ defined on the product space $\mathcal{Y}^j$, is predictive sufficient if, and only if, it is parametric sufficient.*

Proof. See Sec. 4.5 of Bernardo and Smith [15] for a heuristic proof. ■

[5] Unless otherwise noted, for an infinite sequence of random variables we use the phrases *exchangeable* and *infinitely exchangeable* interchangeably.

The following theorem identifies the structure in the probability model that leads to the existence of parametric sufficient statistics, thus providing insight into how to propose and test statistics for such sufficiency.

Theorem 2.3.2 (Neyman factorization criterion). *The sequence of statistics $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots$ is parametric sufficient for an infinitely exchangeable sequence of random variables $y_1, y_2, \ldots$ if and only if the joint density for $y_1, \ldots, y_m$ can be factored as*

$$p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \mid \boldsymbol{\theta}) = h_m(\boldsymbol{t}_m, \boldsymbol{\theta}) g(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m), \qquad m \geq 1, \tag{2.51}$$

for some functions $h_m \geq 0$ and $g > 0$.

Proof. This proof follows that provided in Sec. 4.5 of [15]. Assume such a factorization exists. Then, for any $dQ(\boldsymbol{\theta})$ we may write

$$dQ(\boldsymbol{\theta} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) = \frac{p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \mid \boldsymbol{\theta}) dQ(\boldsymbol{\theta})}{\int_\Theta p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \mid \boldsymbol{\vartheta}) dQ(\boldsymbol{\vartheta})} = \frac{h_m(\boldsymbol{t}_m, \boldsymbol{\theta}) dQ(\boldsymbol{\theta})}{\int_\Theta h_m(\boldsymbol{t}_m, \boldsymbol{\vartheta}) dQ(\boldsymbol{\vartheta})}.$$

The righthand equality depends on the data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ solely through the statistic $\boldsymbol{t}_m$, and thus, $dQ(\boldsymbol{\theta} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) = dQ(\boldsymbol{\theta} \mid \boldsymbol{t}_m)$.

Conversely, assume that $\boldsymbol{t}_m$ is a parametric sufficient statistic. Then,

$$\begin{aligned} \frac{p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \mid \boldsymbol{\theta}) dQ(\boldsymbol{\theta})}{p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)} &= dQ(\boldsymbol{\theta} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) \\ &= dQ(\boldsymbol{\theta} \mid \boldsymbol{t}_m) = \frac{p(\boldsymbol{t}_m \mid \boldsymbol{\theta}) dQ(\boldsymbol{\theta})}{p(\boldsymbol{t}_m)}. \end{aligned}$$

The result follows by identifying that this must imply

$$p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \mid \boldsymbol{\theta}) = h_m(\boldsymbol{t}_m, \boldsymbol{\theta}) g(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$$

for some $h_m \geq 0$, $g > 0$. ∎

From the Neyman factorization criterion, and from the fact that we can write the likelihood of N i.i.d. observations from a k-parameter exponential family as

$$p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \mid \boldsymbol{\theta}) = \exp\left\{ \eta(\boldsymbol{\theta})^T \sum_{n=1}^N \boldsymbol{t}(\boldsymbol{y}_n) - N\alpha(\boldsymbol{\theta}) + \sum_{n=1}^N \beta(\boldsymbol{y}_n) \right\} \tag{2.52}$$

$$= \exp\left\{ \eta(\boldsymbol{\theta})^T \sum_{n=1}^N \boldsymbol{t}(\boldsymbol{y}_n) - N\alpha(\boldsymbol{\theta}) \right\} \exp\left\{ \sum_{n=1}^N \beta(\boldsymbol{y}_n) \right\}, \tag{2.53}$$

we see that $\boldsymbol{s}_n(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \{n, \sum_{i=1}^n t_1(\boldsymbol{y}_i), \ldots, \sum_{i=1}^n t_k(\boldsymbol{y}_i)\}$, $n = 1, 2, \ldots$, is a sequence of sufficient statistics.

Furthermore, the Pitman-Koopman-Darmois theorem [80, 141] states that a probability model admits a sufficient statistic whose dimension remains bounded as the sample size increases if and only if it is an exponential family model. The first proof of this result is due to Darmois [32] (in French), with two versions in English produced independently by Pitman [136] and Koopman [98], each using slightly different technical conditions.

2.4 Incorporating Prior Knowledge

Within the Bayesian framework, motivated by the concepts presented in Sec. 2.1.1, one is interested in incorporating a prior distribution on the latent model parameter θ in order to make predictions about future data. Assuming the associated conditional densities exist, as we will throughout this section, and given N i.i.d. observations, this *predictive likelihood* is given by:

$$p(\boldsymbol{y} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_N, \lambda) = \int_\Theta p(\boldsymbol{y} \mid \boldsymbol{\vartheta}) p(\boldsymbol{\vartheta} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_N, \lambda) d\boldsymbol{\vartheta}. \tag{2.54}$$

Here, we take the prior distribution itself to be contained within a family $\mathcal{P}_\Lambda$ parameterized by a set of *hyperparameters* $\lambda \in \Lambda$. The hyperparameters are not fundamental to the objective of our inference, and can simply be viewed as tuning parameters. As an intermediary step in the process of predictive analysis, one might simply be interested in examining the *posterior density* on $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}, \lambda) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \lambda)}{\int_\Theta p(\boldsymbol{y} \mid \boldsymbol{\vartheta}) p(\boldsymbol{\vartheta} \mid \lambda) d\boldsymbol{\vartheta}} \tag{2.55}$$

There are many perspectives on how one should choose a prior distribution on the latent parameter θ. A *subjective Bayesian* would argue that one should choose a distribution that encodes our subjective prior belief about the values of $\boldsymbol{\theta}$. On the other hand, *objective Bayesians* aim to remain agnostic and employ a prior distribution that is maximally uninformative, allowing the data to speak most loudly. Such goals often lead to the use of "flat" priors (e.g., limiting forms of the conjugate families discussed in Sec. 2.4.1), but with sometimes unintended implications [13]. A more coherent framework for developing objective priors is that of *reference analysis*, first introduced by Bernardo [14] and further developed by Berger and Bernardo [12][6]. A reference prior is one that—constrained within a class of candidate priors—maximizes the uncertainty about $\boldsymbol{\theta}$ relative to the knowledge that could be gained about $\boldsymbol{\theta}$ from repeated observations from the model. For any sufficiently regular prior $p(\boldsymbol{\theta})$, as the number of observations tends to infinity, the posterior of $\boldsymbol{\theta}$ concentrates about its true value. Thus, the limiting mutual information

$$\lim_{k \to \infty} \int_{\mathcal{Y}^k} \int_\Theta p(\boldsymbol{\vartheta}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_k) \log \frac{p(\boldsymbol{\vartheta}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_k)}{p(\boldsymbol{\vartheta}) p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_k)} d\boldsymbol{\vartheta} d\boldsymbol{y}_{1:k}, \tag{2.56}$$

or equivalently, the average divergence between the prior and the posterior:

$$\lim_{k \to \infty} \int_{\mathcal{Y}^k} p(\boldsymbol{y}_1, ..., \boldsymbol{y}_k) \int_\Theta p(\boldsymbol{\vartheta} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_k) \log \frac{p(\boldsymbol{\vartheta} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_k)}{p(\boldsymbol{\vartheta})} d\boldsymbol{\vartheta} d\boldsymbol{y}_{1:k} \tag{2.57}$$

provides a measure of the amount of *missing information* about $\boldsymbol{\theta}$ after receiving infinitely many observations from the model. A reference prior aims to choose from

[6] See also [13] for a comprehensive survey.

within a specified class the prior that maximizes this missing information. Thus, reference priors only depend on the asymptotic behavior of the assumed model[7]. In finite parameter spaces (i.e., $|\Theta| = K, K < \infty$), the reference prior reduces to the prior that maximizes the entropy within the class of candidate priors, as proposed by Jaynes [79]. For one-dimensional *location and scale families*, such as the family of univariate Gaussian distributions parameterized by a mean (location parameter) and variance (scale parameter), the reference prior is a constant for the location parameter (i.e., improper) and equivalent to Jeffreys prior [81] for the scale parameter. For more complex models, however, derivation of reference priors relies on numerical techniques. In addition, even once a reference prior is derived, posterior inference can be challenging.

An alternative approach, largely considered a pragmatic choice, is that of conjugate priors. In many cases, these priors do indeed encode substantial information that can strongly influence the analysis of $\boldsymbol{\theta}$.[8] As we see in the following sections, the parametrization of these conjugate priors can be viewed as adding pseudo-observations when the model class is in the regular exponential family. Thus, choices of hyperparameters that add few pseudo-observations are often viewed (with the caveats mentioned above) as *weakly informative* while maintaining the computational benefits we describe in Sec. 2.4.1. For the subjective Bayesian, the choice of a conjugate prior is also a pragmatic one, and the hyperparameters allow for a simple method of tuning the distribution to aspects of their prior belief.

■ 2.4.1 Conjugate Priors

The use of *conjugate priors* is often motivated by practical considerations. Namely, conjugate priors allow for a computationally tractable mechanism for incorporating new data into the posterior distribution of the parameter $\boldsymbol{\theta}$. For an arbitrary family $\mathcal{P}_\Lambda$ of prior distributions, with $p(\boldsymbol{\theta} \mid \lambda) \in \mathcal{P}_\Lambda$, the integral of Eq. (2.54) and in the denominator of Eq. (2.55) may be intractable. If, however, $p(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta} \mid \lambda)$ remains in the family $\mathcal{P}_\Lambda$ where every element of $\mathcal{P}_\Lambda$ has some known functional form, then the normalization constant is automatically determined by the definition of the distributions in that family. This motivates the following definition of conjugacy.

Definition 2.4.1. *A family $\mathcal{P}_\Lambda$ of prior distributions on $\boldsymbol{\theta} \in \Theta$ is said to be* conjugate *to a model class $\mathcal{P}_\Theta$, with $p(\boldsymbol{y} \mid \boldsymbol{\theta}) \in \mathcal{P}_\Theta$, if the posterior remains in the family of prior distributions:*

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}, \lambda) \in \mathcal{P}_\Lambda \tag{2.58}$$

[7] This statement assumes the model provides conditionally independent observations given $\boldsymbol{\theta}$. If instead there were dependencies in the observations, such as the time-series models we consider in this thesis, the reference prior might be a function of the sample size.

[8] We note, however, that there are special cases in which the conjugate prior and reference prior coincide. For example, these priors coincide when θ represents the mean of a Gaussian, and that mean is subject to second moment constraints. In this case, both the reference and conjugate priors are Gaussian, which can be derived utilizing the constrained maximum entropy results of Sec. 2.2.2 and noting the equivalence of the reference prior and the maximum entropy prior for location families.

for all possible observations $\boldsymbol{y} \in \mathcal{Y}$*, likelihoods* $p(\cdot \mid \boldsymbol{\theta}) \in \mathcal{P}_\Theta$*, and priors* $p(\cdot \mid \lambda) \in \mathcal{P}_\Lambda$.[9]

Since we could simply take $\mathcal{P}_\Lambda$ to be the set of all distributions, this definition alone does not lead to the tractable inference we seek to define. Instead, one may consider the likelihood in terms of a sufficient statistics $\boldsymbol{t}(\cdot)$. From the definition of sufficiency, we have

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}, \lambda) = p(\boldsymbol{\theta} \mid \boldsymbol{t}(\boldsymbol{y}), \lambda) \propto p(\boldsymbol{t}(\boldsymbol{y}) \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \lambda) \tag{2.59}$$

If $\boldsymbol{t}(\boldsymbol{y})$ is of fixed, finite dimension independent of that of $\boldsymbol{y}$ (i.e., the number of data points), the family of prior probability distributions which satisfy

$$p(\boldsymbol{t}(\boldsymbol{y}) \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \lambda) \propto p(\boldsymbol{\theta} \mid \lambda') \tag{2.60}$$

will lead to tractable inference. From this stricter definition, the Pitman-Koopman-Darmois theorem [80] described at the end of Sec. 2.3 implies that the class of likelihoods for which a conjugate prior family exists are those belonging to the exponential family (regular or non-regular).

Conjugate Prior to the Regular Exponential Family

Given a model which is member of a regular exponential family, we may easily construct the corresponding conjugate prior. Namely, for any member of $\mathcal{E}(\boldsymbol{\theta}; \boldsymbol{\eta}(\cdot), \boldsymbol{t}(\cdot), \beta(\cdot))$, we can write the likelihood as:

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \exp\{\boldsymbol{\eta}(\boldsymbol{\theta})^T \boldsymbol{t}(\boldsymbol{y}) - \alpha(\boldsymbol{\theta}) + \beta(\boldsymbol{y})\} \tag{2.61}$$

Given a set of N i.i.d. observations, we have:

$$\begin{aligned} p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \mid \boldsymbol{\theta}) &= \exp\left\{\boldsymbol{\eta}(\boldsymbol{\theta})^T \sum_{n=1}^N \boldsymbol{t}(\boldsymbol{y}_n) - N\alpha(\boldsymbol{\theta}) + \sum_{n=1}^N \beta(\boldsymbol{y}_n)\right\} && (2.62) \\ &= \exp\left\{\boldsymbol{\eta}(\boldsymbol{\theta})^T \sum_{n=1}^N \boldsymbol{t}(\boldsymbol{y}_n) - N\alpha(\boldsymbol{\theta})\right\} \exp\left\{\sum_{n=1}^N \beta(\boldsymbol{y}_n)\right\} && (2.63) \end{aligned}$$

If we choose $\mathcal{P}_\Lambda$ such that

$$\mathcal{P}_\Lambda = \{p(\cdot \mid \lambda) \mid p(\boldsymbol{\theta} \mid \lambda) \propto \exp\{\boldsymbol{t}_0^T \boldsymbol{\eta}(\boldsymbol{\theta}) - N_0 \alpha(\boldsymbol{\theta})\}\} \tag{2.64}$$

with $\lambda = \{\boldsymbol{t}_0, N_0\}$, then

$$\begin{aligned} p(\boldsymbol{\theta} \mid \boldsymbol{y}_1, \ldots \boldsymbol{y}_N, \lambda) &\propto \exp\{\boldsymbol{t}'^T \boldsymbol{\eta}(\boldsymbol{\theta}) - N' \alpha(\boldsymbol{\theta})\} && (2.65) \\ &= p(\boldsymbol{\theta} \mid \lambda') \in \mathcal{P}_\Lambda && (2.66) \end{aligned}$$

[9] Occasionally, we write $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ to be explicit about the domain of the distribution, whereas here we write $p(\cdot \mid \boldsymbol{\theta})$ to be clear that the distribution is a *function* of its argument for fixed $\boldsymbol{\theta}$, not a number resulting from an evaluation at $\boldsymbol{y}$.

with $\lambda' = \{\boldsymbol{t}', N'\}$ where

$$\boldsymbol{t}' = \boldsymbol{t}_0 + \sum_{n=1}^{N} \boldsymbol{t}(\boldsymbol{y}_n) \tag{2.67}$$

$$N' = N_0 + N. \tag{2.68}$$

We note that the conjugate prior is itself in the exponential family. Namely, the prior is in the canonical family $\mathcal{E}(\boldsymbol{t}_0; \boldsymbol{I}(\cdot), \boldsymbol{\eta}(\cdot), -N_0\alpha(\cdot))$. As evidenced by Eq. (2.67)-Eq. (2.68), we see that conjugate priors have the additional benefit of being interpretable as simply adding N_0 pseudo-observations with a total sufficient statistic $\boldsymbol{t}_0$.

The likelihood of the data can then be written in terms of the normalizing constant of a member of the exponential family:

$$\begin{aligned}
p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_N \mid \lambda) &= \int_\Theta p(\boldsymbol{\vartheta} \mid \lambda) p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_N \mid \boldsymbol{\vartheta}) d\boldsymbol{\vartheta} \\
&= \int_\Theta \exp\left\{ \boldsymbol{\eta}(\boldsymbol{\vartheta})^T \sum_{n=1}^{N} \boldsymbol{t}(\boldsymbol{y}_n) - N\alpha(\boldsymbol{\vartheta}) + \sum_{n=1}^{N} \beta(\boldsymbol{y}) \right\} \\
&\qquad \exp\{\boldsymbol{t}_0^T \boldsymbol{\eta}(\boldsymbol{\vartheta}) - \gamma(\lambda) - N_0\alpha(\boldsymbol{\vartheta})\} d\boldsymbol{\vartheta} \\
&= \exp\left\{ -\gamma(\lambda) + \sum_{n=1}^{N} \beta(\boldsymbol{y}) \right\} \exp\{\boldsymbol{t}'^T \boldsymbol{\eta}(\boldsymbol{\vartheta}) - N'\alpha(\boldsymbol{\vartheta})\} d\boldsymbol{\vartheta} \\
&= \exp\left\{ \gamma(\lambda') - \gamma(\lambda) + \sum_{n=1}^{N} \beta(\boldsymbol{y}) \right\},
\end{aligned} \tag{2.69}$$

where we use $\gamma(\cdot)$ to denote the log-partition function of the conjugate prior family P_Λ, and the last equality follows from identifying the integral over Θ as integrating over an unnormalized member of P_Λ with parameter $\lambda' = \{\boldsymbol{t}', N'\}$.

In the following sections, we briefly outline some of the probability density and mass functions, and the associated conjugate analysis that we utilize throughout this thesis. All of these results may be derived using a combination of the results presented in Sec. 2.4.1 along with manipulations similar to those in Sec. 2.2.3.

■ 2.4.2 Multinomial Observations

Multinomial Likelihood Distribution

Consider a random variable y on a finite sample space $\mathcal{Y} = \{1, \dots, K\}$. Let the probability mass function be denoted by $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]$. The *multinomial* distribution [16, 51] describes the probability of a string of N observations of y taking on values $y_1, \dots, y_n$:

$$p(y_1, \dots, y_N | \boldsymbol{\pi}) = \frac{N!}{\prod_k N_k!} \prod_k \pi_k^{N_k}, \quad N_k \triangleq \sum_n \delta(y_n, k). \tag{2.70}$$

We use the notation $\delta(j, k)$ to indicate the discrete Kronecker delta. When $K = 2$, this distribution is referred to as the *binomial* distribution.

Dirichlet Prior Distribution

The K-dimensional *Dirichlet* distribution [51] is the conjugate prior for the class of K-dimensional multinomial distributions and is uniquely defined by a set of hyperparameters $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_K]$. The distribution has the following form:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k - 1}, \quad \alpha_k > 0, \tag{2.71}$$

with $\Gamma(\cdot)$ representing the standard Gamma function. We denote this distribution by $\mathrm{Dir}(\alpha_1, \dots, \alpha_K)$. When $K = 2$, this distribution is referred to as the *beta* distribution, which we denote by $\mathrm{Beta}(\alpha_1, \alpha_2)$. The first moment of the Dirichlet distribution is given by:

$$\mathbb{E}[\pi_i] = \frac{\alpha_i}{\sum_j \alpha_j}. \tag{2.72}$$

Conjugate Posterior and Predictions

The conjugacy of the Dirichlet distribution implies that, conditioned on N multinomial observations $y_1, \dots, y_N$, the posterior distribution of $\boldsymbol{\pi}$ is also Dirichlet:

$$\begin{aligned} p(\boldsymbol{\pi}|y_1, \dots, y_N, \boldsymbol{\alpha}) &\propto p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(y_1, \dots, y_N|\boldsymbol{\pi}) & (2.73) \\ &\propto \prod_{k+1}^{K} \pi_k^{\alpha_k + N_k - 1} \propto \mathrm{Dir}(\alpha_1 + N_1, \dots, \alpha_K + N_K). & (2.74) \end{aligned}$$

Using the normalizing constant of the Dirichlet distribution, and substituting into Eq. (2.54), one can derive the predictive likelihood to be:

$$p(y = k|y_1, \dots, y_N, \boldsymbol{\alpha}) = \frac{N_k + \alpha_k}{N + \alpha_0}, \quad \alpha_0 \triangleq \sum_{k=1}^{K} \alpha_k. \tag{2.75}$$

■ 2.4.3 Gaussian Observations

Gaussian Likelihood Distribution

A *Gaussian* or *normal* distribution [51] is parameterized by a mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. This distribution often arises in the natural world and can provide a useful description of continuous-valued random variables that concentrate about a given value and have constrained variability. The distribution is defined over a sample space $\mathcal{Y} = \mathbb{R}^d$ and is written as

$$p(\boldsymbol{y}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \right\}. \tag{2.76}$$

We denote this Gaussian distribution by $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ or $\mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}, \Sigma)$ to be explicit about the domain.

Known Covariance: Normal Prior Distribution

For fixed covariance Σ, the normal distribution is the conjugate prior on the mean parameter $\boldsymbol{\mu}$. In the following, we assume a $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ prior for this parameter[10].

Known Mean: Inverse-Wishart Prior Distribution

When only the covariance Σ is uncertain, the conjugate prior is the *inverse-Wishart* distribution [51]. The d-dimensional inverse-Wishart distribution, with covariance parameter Δ and ν degrees of freedom, is given by

$$p(\Sigma|\nu, \Delta) = \frac{|\nu\Delta|^{\frac{\nu}{2}} |\Sigma|^{-\frac{\nu+d+1}{2}}}{2^{\frac{\nu d}{2}} \pi^{\frac{d(d-1)}{4}} \prod_{i=1}^{k} \Gamma\left(\frac{\nu+1-i}{2}\right)} \exp\left\{-\frac{1}{2}\mathrm{tr}(\nu\Delta\Sigma^{-1})\right\}. \tag{2.77}$$

We denote this distributions by $\mathrm{IW}(\nu, \Delta)$. The first moment is given by:

$$\mathbb{E}[\Sigma] = \frac{\nu\Delta}{\nu - d - 1}. \tag{2.78}$$

Normal-Inverse-Wishart Prior Distribution

When both the mean and covariance are uncertain, the *normal-inverse-Wishart* distribution [51] is conjugate. This distribution defines a conditionally normal prior on the mean, $\boldsymbol{\mu} \mid \Sigma \sim \mathcal{N}(\boldsymbol{\vartheta}, \Sigma/\kappa)$, and an inverse-Wishart distribution on the covariance, $\Sigma \sim \mathrm{IW}(\nu, \Delta)$. The joint prior distribution is then defined as

$$p(\boldsymbol{\mu}, \Sigma|\kappa, \boldsymbol{\vartheta}, \nu, \Delta) \propto |\Sigma|^{-\frac{\nu+d+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\nu\Delta\Sigma^{-1}) - \frac{\kappa}{2}(\boldsymbol{\mu} - \boldsymbol{\vartheta})^T \Sigma^{-1} (\boldsymbol{\mu} - \boldsymbol{\vartheta})\right\}. \tag{2.79}$$

We will use the notation $\mathcal{NIW}(\kappa, \boldsymbol{\vartheta}, \nu, \Delta)$ to represent this distribution[11].

Conjugate Posteriors and Predictions

Consider N Gaussian observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ with $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. We will outline the posterior distributions for each of the three cases listed above. More explicit details can be found in Gelman et al. [51].

For known covariance Σ, the posterior distribution on the mean $\boldsymbol{\mu}$ is given by an updated normal distribution:

$$\begin{aligned} &p(\boldsymbol{\mu} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N, \Sigma, \boldsymbol{\mu}_0, \Sigma_0) \\ &\quad = \mathcal{N}\left((\Sigma_0^{-1} + \Sigma^{-1})^{-1}(\Sigma_0^{-1}\boldsymbol{\mu}_0 + \Sigma^{-1}\sum_{i=1}^{N} \boldsymbol{y}_i), (\Sigma_0^{-1} + \Sigma^{-1})^{-1}\right). \end{aligned} \tag{2.80}$$

[10] In the limit as prior precision tends to zero (i.e., $|\Sigma_0^{-1}| \to 0$), the reference prior $p(\boldsymbol{\mu}) \propto$ constant is obtained.

[11] In the limit as $\kappa \to 0$, $\nu \to -1$, and $|\Delta| \to 0$, the often proposed "noninformative" multivariate Jeffreys prior is obtained: $p(\boldsymbol{\mu}, \Sigma) \propto |\Sigma|^{-(d+1)/2}$. Note, however, in the multivariate case, this is not equivalent to the reference prior [13].

For known mean $\boldsymbol{\mu}$, the posterior distribution on the covariance Σ is given by an updated inverse-Wishart distribution:

$$p(\Sigma|\boldsymbol{y}_1,\dots,\boldsymbol{y}_N,\boldsymbol{\mu},\nu,\Delta)=\mathrm{IW}\left(\nu+N,\Delta+(1/\nu)\sum_{i=1}^{N}(\boldsymbol{y}_i-\boldsymbol{\mu})(\boldsymbol{y}_i-\boldsymbol{\mu})^T\right). \tag{2.81}$$

Finally, when both the mean $\boldsymbol{\mu}$ and covariance Σ are uncertain, the posterior distribution is given by an updated normal inverse-Wishart distribution:

$$p(\boldsymbol{\mu},\Sigma|\boldsymbol{y}_1,\dots,\boldsymbol{y}_N,\kappa,\vartheta,\nu,\Delta)=\mathcal{NIW}(\bar{\kappa},\bar{\boldsymbol{\vartheta}},\bar{\nu},\bar{\Delta}), \tag{2.82}$$

where the hyperparameter update equations are:

$$\bar{\kappa}=\kappa+N \tag{2.83}$$

$$\bar{\kappa}\bar{\boldsymbol{\vartheta}}=\kappa\boldsymbol{\vartheta}+\sum_{n=1}^{N}\boldsymbol{y}_n \tag{2.84}$$

$$\bar{\nu}=\nu+N \tag{2.85}$$

$$\bar{\nu}\bar{\Delta}=\nu\Delta+\sum_{n=1}^{N}\boldsymbol{y}_n\boldsymbol{y}_n^T+\kappa\boldsymbol{\vartheta}\boldsymbol{\vartheta}^T-\bar{\kappa}\bar{\boldsymbol{\vartheta}}\bar{\boldsymbol{\vartheta}}^T. \tag{2.86}$$

For the scenario where both $\boldsymbol{\mu}$ and Σ are uncertain, and a conjugate normal inverse-Wishart prior is placed on these parameters, the predictive likelihood is given by a multivariate Student-t distribution [51]:

$$p(\boldsymbol{y}\mid\boldsymbol{y}_1,\dots,\boldsymbol{y}_N,\kappa,\vartheta,\nu,\Delta)=t_{\bar{\nu}-d+1}\left(\bar{\boldsymbol{\vartheta}},\frac{(\bar{\kappa}+1)\bar{\nu}}{\bar{\kappa}(\bar{\nu}-d+1)}\bar{\Delta}\right), \tag{2.87}$$

where a standard multivariate Student-t distribution $t_\nu(\boldsymbol{\vartheta},\nu\Delta)$ is given by:

$$p(\boldsymbol{\theta})=\frac{\Gamma((\nu+d)/2)}{\Gamma(\nu/2)\nu^{d/2}\pi^{d/2}}|\nu\Delta|^{-1/2}\left(1+\frac{1}{\nu}(\boldsymbol{\theta}-\boldsymbol{\vartheta})^T(\nu\Delta)^{-1}(\boldsymbol{\theta}-\boldsymbol{\vartheta})\right)^{(\nu+d)/2}. \tag{2.88}$$

When $\bar{\nu}>(d+1)$, the posterior density can be approximated by a moment-matched Gaussian:

$$p(\boldsymbol{y}|\boldsymbol{y}_1,\dots,\boldsymbol{y}_N,\kappa,\boldsymbol{\vartheta},\nu,\Delta)\approx\mathcal{N}\left(\boldsymbol{y};\bar{\boldsymbol{\vartheta}},\frac{(\bar{\kappa}+1)\bar{\nu}}{\bar{\kappa}(\bar{\nu}-d+1)}\bar{\Delta}\right). \tag{2.89}$$

For analysis on the accuracy of this approximation, see [157, Section 2.2].

■ 2.4.4 Multivariate Linear Regression Model

Gaussian Likelihood Distribution

The *normal multivariate linear regression model* is one in which the observations, or *responses*, $\boldsymbol{y}_i\in\mathbb{R}^d$ can be described as a linear combination of a set of known *regressors*

$\boldsymbol{x}_i \in \mathbb{R}^n$ with errors accounted for by additive Gaussian noise:

$$\boldsymbol{y}_i = x_{i1}\boldsymbol{a}_1 + \cdots + x_{in}\boldsymbol{a}_n + \boldsymbol{e}_i \qquad \boldsymbol{e}_i \sim \mathcal{N}(0, \Sigma) \tag{2.90}$$

We may combine a set of N response vectors into a matrix $Y = \begin{bmatrix}\boldsymbol{y}_1 & \cdots & \boldsymbol{y}_N\end{bmatrix}$, the regressors into a matrix $X = \begin{bmatrix}\boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N\end{bmatrix}$, and the noise terms into $E = \begin{bmatrix}\boldsymbol{e}_1 & \cdots & \boldsymbol{e}_N\end{bmatrix}$ and compactly write:

$$Y = AX + E, \tag{2.91}$$

where $A = \begin{bmatrix}\boldsymbol{a}_1 & \cdots & \boldsymbol{a}_N\end{bmatrix}$ is referred to as the *design matrix.*

Known Covariance: Matrix-Normal Prior Distribution

When the noise covariance Σ is know, the conjugate prior on the design matrix A is the *matrix-normal distribution* [183]. A matrix $A \in \mathbb{R}^{d \times m}$ has a matrix-normal distribution $\mathcal{MN}(A; M, V, K)$ if

$$p(A) = \frac{|K|^{\frac{d}{2}}}{|2\pi V|^{\frac{m}{2}}} e^{-\frac{1}{2}tr((A-M)^T V^{-1}(A-M)K)}, \tag{2.92}$$

Equivalently,

$$p(\text{vec}(A)) = \mathcal{N}(\text{vec}(M), K^{-1} \otimes V), \tag{2.93}$$

where $\otimes$ denotes the Kronecker product. From this, we see that M is the mean matrix, and V and K^{-1} are related to the covariance along the rows and columns of A.

Matrix-Normal Inverse-Wishart Prior Distribution

The conjugate prior on the set of parameters A and Σ is the *matrix-normal inverse-Wishart* prior. This distribution places a conditionally matrix-normal prior on A given Σ,

$$A \mid \Sigma \sim \mathcal{MN}(A; M, K, \Sigma) \tag{2.94}$$

and an inverse-Wishart prior on Σ,

$$\Sigma \sim \text{IW}(\nu, \Delta). \tag{2.95}$$

Conjugate Posteriors and Predictions

Let $D = \{X, Y\}$. The posterior distribution of $\{A, \Sigma\}$ decomposes as

$$p(A, \Sigma \mid D) = p(A \mid \Sigma, D)p(\Sigma \mid D). \tag{2.96}$$

The resulting posterior of A is derived in Appendix F.1 to be

$$p(A \mid \Sigma, D) = \mathcal{MN}\left(A; S_{yx}S_{xx}^{-1}, \Sigma^{-1}, S_{xx}\right), \tag{2.97}$$

with

$$S_{xx} = XX^T + K \qquad S_{yx} = YX^T + MK \qquad S_{yy} = YY^T + MKM^T. \tag{2.98}$$

The marginal posterior of Σ is given by:

$$p(\Sigma \mid D) = \mathrm{IW}\left(\nu + N, \Delta + S_{y|x}\right), \tag{2.99}$$

where $S_{y|x} = S_{yy} - S_{yx}S_{xx}^{-1}S_{yx}^T$.

■ 2.5 Graphical Models

Probabilistic *graphical models* provide a framework for compactly encoding the conditional probabilistic dependency structure of a set of random variables. For surveys of these models and their associated inference algorithms, see [85, 103, 157, 176], with seminal work by Pearl [134]. The framework of graphical models has allowed for the development of many efficient inference techniques such as *belief propagation* [104, 134], and for advances in *variational methods* [176]. Such developments have provided an ability to analyze large-scale datasets, which would not be feasible without harnessing the sparsity in the parametrization of the full model. Additionally, the generic formulation of the graphical model inference algorithms enables transfer of advances in one domain to other domains in a straightforward manner. For example, many classical models such as the *hidden Markov model* (HMM) [139] and *state space model* can be formulated within the graphical model framework; the inference algorithms developed specifically for these models—like the *forward-backward algorithm* [139], *Viterbi decoding* [42], and *Kalman filtering* [90]—can be derived as special cases of generic graphical model inference algorithms. The development of inference algorithms for the Bayesian nonparametric extensions of these models that we examine in this thesis is considerably simplified by representing the models within the graphical model framework.

■ 2.5.1 A Brief Overview

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of *nodes* $\mathcal{V}$ representing the random variables of the model and *edges* $\mathcal{E}$ containing elements $(i,j) \in \mathcal{E}$ which connect a unique pair of nodes $i, j \in \mathcal{V}$. For an *undirected graph*, the element $(i,j) \in \mathcal{E}$ if and only if $(j,i) \in \mathcal{E}$, which of course need not be true in a *directed graph*. Pictorially, a node is typically represented by a circle, an undirected edge by a line, and a directed edge by an arrow with the tail originating at the *parent* node and the head ending at the *child* node. See Fig. 2.5. We primarily restrict our attention to directed graphs, since these graphs are the most appropriate for describing the dynamical models we consider in this thesis.

■ 2.5.2 Directed Graphical Models

Let $\Gamma(j)$ denote the set of parent nodes to a node i. This set is defined by

$$\Gamma(j) = \{i \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}. \tag{2.100}$$

A *leaf* node is one that has no children while a *root* node has no parents. For a directed graph, the joint density decomposes as the product of the conditional densities for each node i given its parents $\Gamma(i)$:

$$p(\boldsymbol{x}_{\mathcal{V}}) = \prod_{i \in \mathcal{V}} p(x_i \mid \boldsymbol{x}_{\Gamma(i)}), \tag{2.101}$$

where we use the notation $\boldsymbol{x}_{\mathcal{A}}$ to denote the set $\{x_i \mid i \in \mathcal{A}\}$. For an *acyclic* graph (i.e., one without a directed cycle going from some node i and returning to node i,) one can verify that Eq. (2.101) defines a valid joint density. Namely, to verify that the density integrates to 1, one can marginalize over nodes starting at leaf nodes and ending at root nodes. For a directed graph, the sparsity of the model parametrization is defined in terms of the relative ratio of nodes to parent nodes. As we will see, there is not as significant a reduction in the representational complexity and computational complexity of inference if each node has many parents.

Whereas the joint distribution is easy to define from a directed graphical model, the conditional independence statements encoded by the graph are somewhat challenging to directly infer. Consider the graphical model of Fig. 2.2(d). Without conditioning on y, random variables x and z are independent:

$$p(x, y, z) = p(x)p(z)p(y \mid x, z) \Rightarrow p(x, z) = p(x)p(z). \tag{2.102}$$

However, these variables are not conditionally independent given y:

$$\begin{aligned} p(x, z|y) &\propto p(x, y, z) = p(x)p(z)p(y \mid x, z) \\ &\neq p(x \mid y)p(z \mid y). \end{aligned} \tag{2.103}$$

This phenomenon is referred to as *explaining away*, *Berkson's paradox*, or *selection bias*. For example, imagine that x represented whether or not an earthquake occurred, z whether a burglar is trying to get into the car, and y the car alarm. Earthquakes and car robberies might be independent *a priori*, but upon conditioning on the car alarm being triggered, an increase in the probability of an earthquake results in a decrease in the probability of a burglary since the earthquake "explains away" the fact that the alarm was triggered. For general directed graphical models, instead of writing down the joint distribution and deriving whether the conditional independence statement is true, one can employ an algorithm called *Bayes ball* [150]. One can use such an algorithm to verify the conditional independence statements that appear in the derivations in the appendices of this thesis. The algorithm provides a set of eight scenarios, depicted in Fig. 2.2, consisting of the eight possible three-node chains one can encounter in a

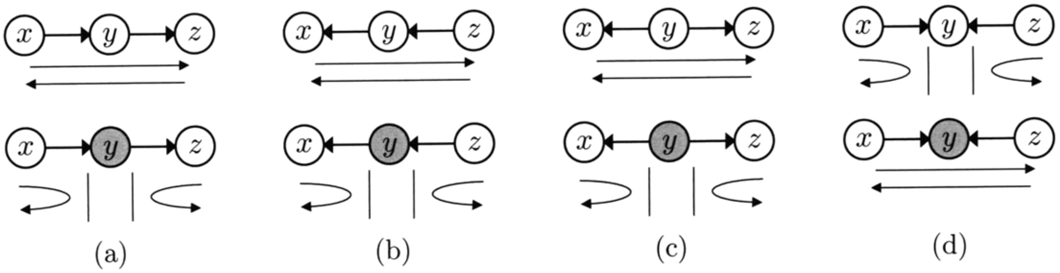


Figure 2.2. Pictorial representation of the Bayes ball algorithm for determining the independence statements in a directed graphical model. There are four possible three node combinations depicted by the graphs of (a)-(d). For each of these structures, we examine the case of marginal independence of x and z (top) or conditional independence of x and z (bottom) given an observation y (gray node). If a ball starting at one of the x or z nodes can pass to the other, as indicated by the straight arrows, then those two nodes are not (conditionally/marginally) independent. If the ball bounces back, as indicated by a set of walls and curved arrows, then the nodes are (conditionally/marginally) independent. These rules can be linked together in various combinations to examine larger graphical models.

directed graph based on directionality of the edges and whether or not the intermediary node is an evidence node (i.e., observed). Some of the junction scenarios are bestowed with a set of walls that deflect the Bayes ball. Two random variables x_i and x_j associated with nodes i and j are then deemed conditionally *dependent* given the random variables $\boldsymbol{x}_{\mathcal{V}_k}$ associated with a set of evidence nodes $\mathcal{V}_k$ (which may be the empty set) if a ball starting at one node can traverse the graph to the other node based on the rules summarized in Fig. 2.2; the random variables are conditionally *independent* otherwise. Another method of determining some statements of conditional independence, and ones extremely useful for the inference algorithms we develop, is described in the following.

Markov Blanket

For a directed graph, a node is conditionally independent of all other nodes in the graph given its *Markov blanket* which consists of the node's parents, children, and *coparents*. The coparents of a given node are defined as those nodes that have a child in common with the given node. The Markov blanket concept is depicted in Fig. 2.3.

Mixture Models and Exchangeability

The version of the de Finetti theorem in Corollary 2.1.1, assuming the distribution Q has a parameterized density $q(\cdot \mid \lambda)$, implies the following hierarchical Bayesian model:

$$p(y_1, \ldots, y_n, \theta \mid \lambda) = q(\theta \mid \lambda) \prod_{i=1}^{n} p(y_i \mid \theta), \tag{2.104}$$

which, based on Eq. (2.101), has a directed graphical representation shown in Fig. 2.4. This figure contains both an explicit representation of the graphical model, as well as an equivalent representation using *plate notation* to compactly represent the n observations

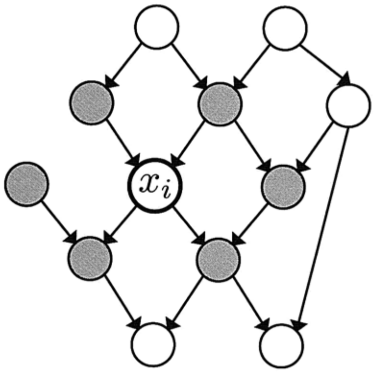


Figure 2.3. Markov blanket for x_t consisting of the node's parents, coparents, and children. The node x_t is then conditionally independent of all other nodes in the graph given its Markov blanket.

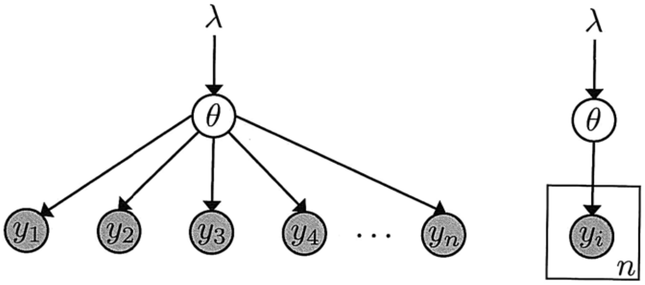


Figure 2.4. Graphical representation of the hierarchical Bayesian model of n exchangeable random variables implied by de Finetti's theorem. Each observation is an independent draw from a density parameterized by θ, which itself has a prior distribution with hyperparameters λ. *Left*: An explicit representation of the graphical model. *Right*: A compact representation using a plate to denote n replicates of the observations y_i.

y_i. The fact that this set of random variables is yielded conditionally i.i.d. given θ can be directly verified from the graphical model by using the Markov blanket concept or the Bayes ball algorithm.

■ 2.5.3 Undirected Graphical Models

Many inference algorithms for directed graphical models rely on first converting the graph to an undirected form. This conversion process, referred to as *moralization*, "marries" any coparents by connecting them with an undirected edge. Each directed edge is then converted into an undirected edge. See Fig. 2.5. In the following, we provide a very brief sketch of the theory of undirected graphical models that we employ in subsequent sections.

Undirected graphical models, or *Markov random fields* (MRF), are typically used when there is no causal structure to the data, as in images, which instead have spatial dependencies. Whereas the directed graphical model is easily derived from the factor-

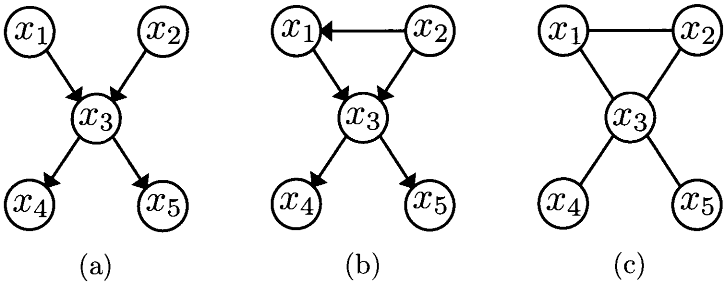


Figure 2.5. (a)-(b) Two directed graphical models that result in the same moralized (undirected) graphical model shown in (c).

ization of the joint distribution, the form of an undirected model is typically formed from a set of conditional independence statements. For an undirected graphical model, if $\mathcal{V}_i$, $\mathcal{V}_j$ and $\mathcal{V}_k$ are three disjoint sets of nodes, and if every path from a node in $\mathcal{V}_i$ to a node in $\mathcal{V}_k$ passes through $\mathcal{V}_j$, then $\mathcal{V}_j$ is called a *separator*. If the following holds for every possible choice of such sets:

$$p(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_k} \mid \boldsymbol{x}_{\mathcal{V}_j}) = p(\boldsymbol{x}_{\mathcal{V}_i} \mid \boldsymbol{x}_{\mathcal{V}_j}) p(\boldsymbol{x}_{\mathcal{V}_k}, \mid \boldsymbol{x}_{\mathcal{V}_j}), \tag{2.105}$$

then the set of random variables $\boldsymbol{x}_{\mathcal{V}} = \{x_i\}$ is said to be *globally Markov* with respect to the undirected graph $\mathcal{G}$. Eq. (2.105) implies that each node i in an undirected graphical model is conditionally independent of all other nodes given its set of *neighbors* $\Gamma(i)$:

$$p(x_i \mid \boldsymbol{x}_{\mathcal{V} \backslash i}) = p(x_i \mid \boldsymbol{x}_{\Gamma(i)}), \tag{2.106}$$

where $\mathcal{V} \backslash i$ denotes the set of all nodes except for node i, and $\Gamma(i)$ is defined just as in Eq. (2.100) using the undirected set of edges $\mathcal{E}$. This *local Markov property* can be used to derive the Markov blanket property of a directed graph since the neighborhood of a node in a moralized graph will solely contain the children, parents, and coparents of the node in the directed graph.

It is important to note that in the conversion of a directed graph to its undirected form, all of the conditional independence statements of the undirected graph hold for the model of the directed graph. The converse is not necessarily true since the mapping is many to one. Take, for example, the graphs of Fig. 2.5. The directed graph of Fig. 2.5(a) encodes a model with x_1 and x_2 marginally independent; however, in the moralized graph this result cannot be directly deduced from the graphical model and instead depends upon the parametrization. For example, the directed graph of Fig. 2.5(c) does not (necessarily) have x_1 and x_2 independent and has the same undirected graphical representation. For the basic V-structure of Fig. 2.2(d), there is no undirected graph that encodes the same set of independence statements. Conversely, for an undirected graph consisting of four nodes x_1, x_2, x_3, x_4 connected in a four-cycle (i.e., each node shares an edge with exactly two other nodes), none of the 16 possible directed graph structures capture the same conditional independence statements. For tree-structured directed graphical models, in which the moralized graph does not contain any loops, the set of conditional independence statements for both graphs is *identical*, implying that undirected graph inference exploits all possible conditional independencies. In Sec. 2.5.4, we present an efficient inference algorithm for undirected, tree-structured graphical models that harnesses the conditional independence statements implied by the graphical model. Because these statements for a moralized directed tree are the same as those for the directed tree, the undirected inference is equivalent to inference in the directed tree and leverages all possible efficiencies. The majority of algorithms developed in this thesis simplify to iterative inferences on tree-structured graphs.

Given a general undirected graph $\mathcal{G}$, the characterization of a joint distribution satisfying the specified Markov properties is not as straightforward as in the directed case. However, the *Hammersley-Clifford theorem* [21] provides some insight. Let $\mathcal{C}$ denote the set of cliques in an undirected graph $\mathcal{G}$, where a *clique* is defined as a fully connected subset of nodes. If a distribution can be factorized in terms of non-negative *potential functions* $\psi_c(\cdot)$ defined on the cliques:

$$p(\boldsymbol{x}_{\mathcal{V}}) \propto \prod_{c \in \mathcal{C}} \psi_c(\boldsymbol{x}_c), \tag{2.107}$$

then the distribution is Markov with respect to $\mathcal{G}$. Conversely, any strictly positive density, $p(x) > 0$ for all x, which is Markov with respect to $\mathcal{G}$ has such a factorized representation. Note that the full characterization of the joint distribution, a necessary step for many inference tasks, can be quite challenging since it relies on computing a normalization constant or *partition function* from an arbitrarily complicated product of potential functions.

In some applications, it is useful to examine a *pairwise Markov random field* representation in which the clique potentials are defined on the graph's edges:

$$p(\boldsymbol{x}_{\mathcal{V}}) \propto \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i \in \mathcal{V}} \psi_k(x_i). \tag{2.108}$$

■ 2.5.4 Belief Propagation

For most graphical models we encounter in applications, the joint state space $\mathcal{X}$ is too large to explicitly characterize, and thus simple inference tasks can pose significant challenges. For example, consider a graphical model with N nodes each taking one of K possible values. The joint state space of such a graph is $|\mathcal{X}| = K^N$. Naive computation of the *posterior marginal* based on a set of observations $\boldsymbol{y}$,

$$p(x_i \mid \boldsymbol{y}) = \int_{\mathcal{X}_{\mathcal{V}\backslash i}} p(\boldsymbol{x}_{\mathcal{V}} \mid \boldsymbol{y}) dx_{\mathcal{V}\backslash i}, \tag{2.109}$$

requires a sum containing K^{N-1} terms in the case of the K-valued graphical model.

For tree-structured graphical models, however, such global inference tasks can be *exactly* and efficiently computed by a recursion of local computations. This *belief propagation* algorithm harnesses the fact that in an undirected tree (such as the one depicted

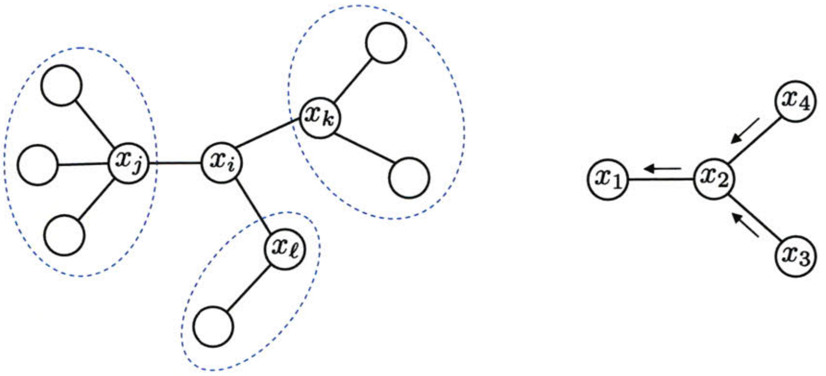


Figure 2.6. *Left:* A tree graphical model with node x_i dividing the tree into disjoint subgraphs. *Right:* A simple tree graph for illustrating the concepts underlying belief propagation.

in Fig. 2.6(left)), any given node separates the tree into disjoint—and thus conditionally independent—subgraphs given the value on the separating node. Computations performed within the subgraphs can then be combined to form the desired posterior marginal of the chosen node.

For the graph of Fig. 2.6(right), the joint distribution can be factorized as follows:

$$p(\boldsymbol{x}) \propto \psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)\psi_{24}(x_2, x_4)\psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\psi_4(x_4). \tag{2.110}$$

Then, computation of the marginal $p(x_1)$ can be accomplished by combining local computations resulting from distributing the integrals over the terms of the product in Eq. (2.110):

$$\begin{aligned} p(x_1) \propto \psi_1(x_1) \int_{\mathcal{X}_2} & \psi_{12}(x_1, x_2)\psi_2(x_2) \\ & \underbrace{\left[\int_{\mathcal{X}_3} \psi_{23}(x_2, x_3)\psi_3(x_3)dx_3\right]}_{m_{32}(x_2)} \cdot \underbrace{\left[\int_{\mathcal{X}_4} \psi_{24}(x_2, x_4)\psi_4(x_4)dx_4\right]}_{m_{42}(x_2)} dx_2. \end{aligned} \tag{2.111}$$

Here, we have defined a *message* $m_{ij}(x_j)$ as the result of a local integration over x_i that results in a function in terms of x_j. In the above example, we would also define

$$m_{21}(x_1) \propto \int_{\mathcal{X}_2} \psi_{12}(x_1, x_2)\psi_2(x_2)m_{32}(x_2)m_{42}(x_2)dx_2. \tag{2.112}$$

More generally, assume we additionally have a set of *evidence nodes* representing a set of observations $\boldsymbol{y} = \{y_i\}$ that are conditioned upon during inference. We assume a structure in which the neighborhood associated with each observation y_i solely contains node i (i.e., that of x_i), and define a generic message from node j to node i as

$$m_{ji}(x_i) = \int_{\mathcal{X}_j} \left(\psi_i(x_i, y_i)\psi_{ij}(x_i, x_j) \prod_{k \in \Gamma(j) \setminus i} m_{kj}(x_j) \right) dx_j. \tag{2.113}$$

That is, each outgoing message from node j is a function of $|\Gamma(j)| - 1$ incoming messages to node j. The initial messages at leaf nodes are simply given by:

$$m_{ji}(x_i) = \int_{\mathcal{X}_j} \psi_i(x_i, y_i)\psi_{ij}(x_i, x_j) dx_j. \tag{2.114}$$

Then, one can show that after passing all of the messages, the desired marginal can be computed as

$$p(x_i \mid \boldsymbol{y}) = \frac{1}{Z}\psi_i(x_i, y_i) \prod_{j \in \Gamma(i)} m_{ji}(x_i), \tag{2.115}$$

with

$$Z = \int_{\mathcal{X}_i} \psi_i(x_i, y_i) \prod_{j \in \Gamma(i)} m_{ji}(x_i) dx_i. \tag{2.116}$$

See [157] for a more complete derivation, and for references to classical literature. Note that tractable propagation of messages and computation of the normalization constant in Eq. (2.116) relies on restricted forms such as discrete or Gaussian MRFs. Otherwise, one can consider a discretization of the continuous beliefs or one of many approximate inference schemes such as the Monte Carlo techniques we outline in Sec. 2.8.

A node can send a valid message to a neighboring node only when it has received valid messages from each of its other neighbors. As such, one needs to implement a *schedule* when running belief propagation. One possible choice is a serial scheme in which a single node is selected as the root of the tree. Then, messages are passed from the leaves to the root, followed by a pass from the root back to the leaves. See Fig. 2.7(top). Alternatively, one can use a synchronous parallel update where every node sends a message whenever it has received all $|\Gamma(j)| - 1$ incoming messages. This schedule starts with all leaf node passing messages, as depicted in Fig. 2.7(bottom). Finally, a parallel scheme involving message passing from all nodes at every iteration is also provably correct. After L such iterations, the local marginal estimates will have incorporated information from all nodes within a distance L [2]. Thus, the algorithm converges after a number of iterations equal to the *diameter* of the tree. Typically, the messages are initialized to be uniform over $\mathcal{X}_i$ in the case of a discrete-valued MRF. The parallel scheme has obvious advantages over the alternative schedules in a distributed implementation; in a serial implementation, such a schedule is typically inefficient but is simple to code.

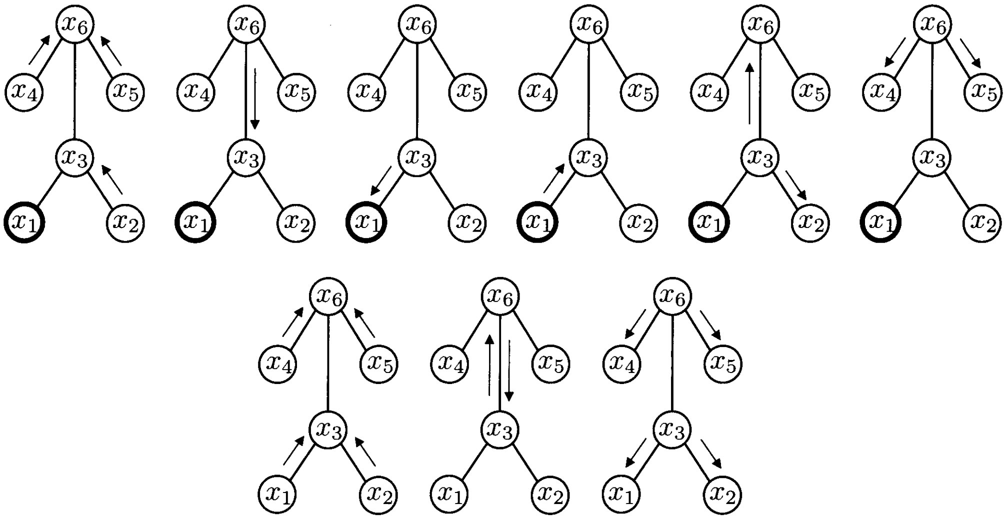


Figure 2.7. Graphical representation of the serial (top) and synchronous (bottom) belief propagation scheduling schemes. Arrows indicate a message being passed. For the serial schedule, the node x_1 was selected as the root node.

In terms of computational costs, the belief propagation algorithm can lead to a dramatic improvement over alternative approaches for computing marginals at *all* nodes. For example, if each node can take on K possible values, and we have N nodes in total, a brute force approach to calculating the set of marginals requires $O(NK^{N-1})$ operations (K^{N-1} operations for computing the marginal at each of N nodes.) A naive application of simply passing integrals through the product of the pairwise potentials in Eq. (2.108) requires $O(N(N-1)K^2)$ operations (for each of N nodes, integrate over $N-1$ nodes with two nodes per clique.) The belief propagation algorithm simply requires $O(NK^2)$ operations by efficiently reusing messages.

Note that for graphs with cycles, a single node does not necessarily partition the graph into disjoint sets, and thus Eq. (2.115) is not valid. The *junction tree algorithm* [104, 151] allows for exact inference in arbitrary graphs by running belief propagation on the tree formed from the maximal cliques of a *triangulated* graph. However, these cliques can be quite large, leading to computation intractability. In such cases, the parallel message update form of belief propagation algorithm is often applied directly to graphs with loops, and is termed *loopy belief propagation.* For graphs with large loops, the inconsistencies or *frustrations* that can arise in more tightly coupled loops are less pronounced and loopy belief propagation can yield good performance. In the Gaussian MRF case, if loopy belief propagation converges, then it provides correct node means (but in general gives incorrect node variances) [181]. For convergence results in discrete and Gaussian MRFs, see [71, 113] for more details.

Many classical models, such as the hidden Markov model (HMM) or linear-Gaussian state space model, have hand-tailored inference algorithms, such as the *forward-backward*

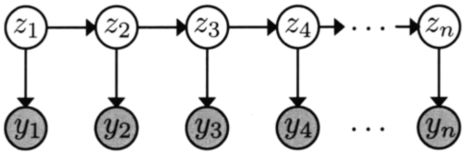


Figure 2.8. Graphical representation of a hidden Markov model (HMM) over n time steps. The latent, discrete-valued Markov process z_t captures the temporal dependencies in the observations y_t.

algorithm and *Kalman filtering and smoothing*, that can be described within the more general framework of inference on a graphical model. In Sec. 2.6-2.7, we examine the HMM and state space models in detail and explore these connections in inference algorithms.

■ 2.6 Hidden Markov Model

The hidden Markov model, or *HMM*, is a class of doubly stochastic processes based on an underlying, discrete-valued state sequence that is modeled as Markovian [139]. Conditioned on this state sequence, the model assumes that the observations, which may be discrete or continuous valued, are independent. The HMM has proven a powerful model in many applied fields including speech recognition [82, 88, 139], computational biology [100, 101, 155], machine translation [127, 128], cryptanalysis [92] and finance [17].

Let z_t denote the *state* of the Markov chain at time t and π_j the state-specific *transition distribution* for state j. Then, the Markovian structure on the state sequence dictates that for all $t > 1$

$$z_t \mid z_{t-1} \sim \pi_{z_{t-1}} \tag{2.117}$$

The state at the first time step is distributed according to an *initial transition distribution* π^0:

$$z_1 \sim \pi^0. \tag{2.118}$$

Given the state z_t, the observation y_t is conditionally independent of the observations and states at other time steps. The observation is simply generated as

$$y_t \mid z_t \sim F(\theta_{z_t}) \tag{2.119}$$

for an indexed family of distributions $F(\cdot)$ where θ_i are the *emission parameters* for state i. Assuming there exists a density associated with $F(\cdot)$, the resulting joint density for n observations is then given by:

$$p(z_{1:n}, y_{1:n}) = \pi^0(z_1)p(y_1 \mid z_1)\prod_{t=2}^{n} p(z_t \mid z_{t-1})p(y_t \mid z_t), \tag{2.120}$$

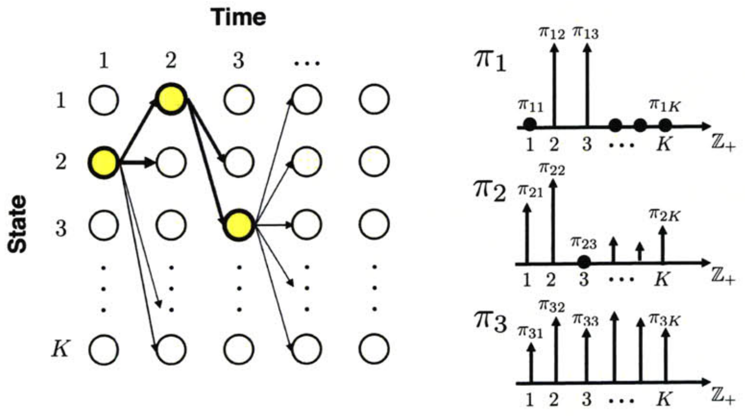


Figure 2.9. *Left:* Lattice representation of an HMM state sequence sample path. Each circle represents one of the K possible HMM states at various time steps. The highlighted circles indicate the selected states, and the arrows represent the set of possible transitions from that state to each of the K possible next states. The weights of these arrows convey the relative probability of the transitions encoded by that state-specific transition distributions π_j. *Right:* Corresponding transition distributions π_1, π_2, and π_3 for the lattice example.

from which we can infer a directed graphical model representation shown in Fig. 2.8. One can use the Bayes ball algorithm of Fig. 2.2 on this HMM graphical model to verify that an observation y_t is indeed conditionally independent of all other observations and states when given the state z_t.

One can view a sample path of the state sequence as a walk through a state versus time lattice, such as the one depicted in Fig. 2.9. A similar diagram representing all possible transitions is often referred to as a *trellis diagram.*

■ 2.6.1 Forward-Backward Algorithm

The *forward-backward algorithm* [139] provides an efficient message-passing scheme for computing node marginals of interest for problems of *filtering* $p(z_n \mid y_1, \dots, y_n)$, *prediction* $p(z_{n+m} \mid y_1, \dots, y_n)$, and *smoothing* $p(z_n \mid y_1, \dots, y_N)$, $N > n$. This classical algorithm has straightforward connections with the belief propagation algorithm of Sec. 2.5.4. Following Rabiner [139], we define a set of *forward messages*

$$\alpha_n(z_n) \triangleq p(y_1, \dots, y_n, z_n) \tag{2.121}$$

and *backward messages*

$$\beta_n(z_n) \triangleq p(y_{n+1}, \dots, y_N \mid z_n). \tag{2.122}$$

For the problem of filtering, we simply need the forward messages since

$$p(z_n \mid y_1, \dots, y_n) = \frac{\alpha_n(z_n)}{\sum_z \alpha_n(z)}. \tag{2.123}$$

Similarly, for prediction we can utilize the Markov structure of the underlying chain to derive that

$$p(z_{n+m} \mid y_1, \ldots, y_n) = \frac{\sum_{z_{n+m-1}} p(z_{n+m} \mid z_{n+m-1}) \cdots \sum_{z_n} p(z_{n+1} \mid z_n) \alpha_n(z_n)}{\sum_z \alpha_n(z)}, \quad (2.124)$$

which we show is equivalent to propagating the forward message without incorporating the missing observations $y_{n+1}, \ldots, y_{n+m}$. The problem of smoothing, on the other hand, utilizes both the forward and backward messages:

$$\begin{aligned} p(z_n \mid y_1, \ldots, y_N) &= \frac{p(y_1, \ldots, y_N \mid z_n) p(z_n)}{p(y_1, \ldots, y_N)} && (2.125) \\ &= \frac{p(y_1, \ldots, y_n \mid z_n) p(y_{n+1}, \ldots, y_N \mid z_n) p(z_n)}{p(y_1, \ldots, y_N)} && (2.126) \\ &= \frac{\alpha_n(z_n) \beta_n(z_n)}{\sum_z \alpha_m(z) \beta_m(z)}, && (2.127) \end{aligned}$$

for any m.

We derive the recursions for these forward and backward messages by harnessing the conditional independencies implied by the graph of Fig. 2.8. The recursions presented in this section are utilized by many of the inference algorithms described throughout the thesis and derived in the appendices. For the forward message,

$$\begin{aligned} \alpha_{n+1}(z_{n+1}) &= p(y_{n+1} \mid z_{n+1}) \sum_{z_n} p(y_1, \ldots, y_n \mid z_n) p(z_{n+1} \mid z_n) p(z_n) && (2.128) \\ &= p(y_{n+1} \mid z_{n+1}) \sum_{z_n} \alpha_n(z_n) p(z_{n+1} \mid z_n). && (2.129) \end{aligned}$$

The backward recursion is similarly derived as

$$\begin{aligned} \beta_n(z_n) &= \sum_{z_{n+1}} p(y_{n+1} \mid z_{n+1}) p(y_{n+2}, \ldots, y_N \mid z_{n+1}) p(z_{n+1} \mid z_n) && (2.130) \\ &= \sum_{z_{n+1}} p(y_{n+1} \mid z_{n+1}) p(z_{n+1} \mid z_n) \beta_{n+1}(z_{n+1}). && (2.131) \end{aligned}$$

The forward initial condition and backward final condition are given by:

$$\begin{aligned} \alpha_1(z_1) &= p(y_1, z_1) = p(y_1 \mid z_1) \pi^0(z_1) && (2.132) \\ \beta_N(z_N) &= 1. && (2.133) \end{aligned}$$

To relate the forward-backward algorithm to belief propagation, we need to convert the directed graph of Fig. 2.8 to its undirected form. In this case, the conversion simply means exchanging directed edges for undirected ones. The relationship between the algorithms is then readily apparent. Specifically, consider node z_n with neighbors z_{n-1}

and z_{n+1}, and evidence node y_n. Choosing a sequential node ordering starting at z_1 for the message-passing scheme, Eq. (2.115) simplifies to:

$$p(z_n \mid y_1, \ldots, y_N) = \frac{1}{Z} \sum_{z_n} p(y_n \mid z_n) m_{n-1,n}(z_n) m_{n+1,n}(z_n), \tag{2.134}$$

with

$$m_{n-1,n}(z_n) = \sum_{z_{n-1}} p(y_{n-1} \mid z_{n-1}) p(z_n \mid z_{n-1}) m_{n-2,n-1}(z_{n-1}) \tag{2.135}$$

$$m_{n+1,n}(z_n) = \sum_{z_{n+1}} p(y_{n+1} \mid z_{n+1}) p(z_{n+1} \mid z_n) m_{n+2,n+1}(z_{n+1}). \tag{2.136}$$

From the above, we can make the connection:

$$\begin{aligned} \alpha_n(z_n) &= p(y_n \mid z_n) m_{n-1,n}(z_n) \\ \beta_n(z_n) &= m_{n+1,n}(z_n). \end{aligned} \tag{2.137}$$

The prediction algorithm of Eq. (2.124) is trivial to derive within the belief propagation framework. Consider an HMM graphical model up to time n and then append a length m Markov chain with nodes $x_{n+1}, \ldots, x_{n+m}$. The standard belief propagation algorithm defined on this graph is then equivalent to the method described above.

■ 2.6.2 Viterbi Algorithm

Given a set of HMM parameters, one might be curious about the most likely state sequence to have generated an observation sequence $y_1, \ldots, y_N$. The *Viterbi algorithm* [42] provides an efficient dynamic programming approach to computing this MAP sequence:

$$\begin{aligned} \hat{\boldsymbol{z}} &= \max_{z_1,\ldots,z_N} \pi^0(z_1) p(y_1 \mid z_1) \prod_{n=2}^{N} p(z_n \mid z_{n-1}) p(y_n \mid z_n) \qquad (2.138) \\ &= \min_{z_1,\ldots,z_N} \left[-\log \pi^0(z_1) - \log p(y_1 \mid z_1) + \sum_{n=2}^{N} -\log p(z_n \mid z_{n-1}) - \log p(y_n \mid z_n) \right]. \end{aligned}$$

Note that choosing the MAP *sequence* is not necessarily equivalent to choosing the maximum node marginal independently at each node:

$$\hat{z}_n = \max p(z_n \mid y_1, \ldots, y_N). \tag{2.139}$$

Actually, such a maximum node marginal sequence may not even be a feasible sequence for the HMM.

The Viterbi algorithm works on the dynamic programming principle that the minimum cost path to $z_n = k$ is equivalent to the minimum cost path to node z_{n-1} plus the cost of a transition from z_{n-1} to $z_n = k$ (and the cost incurred by observation y_n

Compute the MAP hidden Markov model state sequence $\hat{z}_1, \ldots, \hat{z}_N$ as follows:

1. Initialize minimum path sum to state $z_1 = k$ for each $k \in \{1, \ldots, K\}$:
$$\mathcal{S}_1(z_1 = k) = -\log \pi^0(z_1 = k) - \log p(y_1 \mid z_1 = k)$$

2. For $n = 2, \ldots, N$, and for each $k \in \{1, \ldots, K\}$, calculate the minimum path sum to state $z_n = k$:
$$\mathcal{S}_n(z_n = k) = -\log p(y_n \mid z_n = k) + \min_{z_{n-1}} \{\mathcal{S}_{n-1}(z_{n-1}) - \log p(z_n = k \mid z_{n-1})\}$$
and let
$$z^*_{n-1}(z_n) = \arg\min_{z_{n-1}} \{\mathcal{S}_{n-1}(z_{n-1}) - \log p(z_n = k \mid z_{n-1})\}$$

3. Compute
$$\min_{z_1, \ldots, z_N} - \log p(z_1, \ldots, z_N \mid y_1, \ldots, y_N) = \min_{z_N} \mathcal{S}_N(z_N)$$
and set
$$\hat{z}_N = \arg\min_{z_N} \mathcal{S}_N(z_N)$$

4. Iteratively set, for $n \in \{N - 1, \ldots, 1\}$.
$$\hat{z}_n = z^*_n(\hat{z}_{n+1})$$

Algorithm 1. Viterbi hidden Markov model decoding.

given $z_n = k$.) These costs can be represented on edges and nodes in the trellis diagram of Fig. 2.9. The MAP state sequence is then determined starting at node z_N and reconstructing the optimal path backwards in the trellis based on the stored calculations. The details of the Viterbi algorithm are outlined in Algorithm 1.

Viterbi decoding reduces the computation cost to $O(K^2N)$ operations instead of the brute force $O(K^N)$ operations. Algebraically, the Viterbi algorithm is very closely related to the max-product (or min-sum) algorithm that operates by distributing the maximization (or minimization) operators over the elements of the product (or sum) in Eq. (2.138). The max-product algorithm is equivalent to the belief propagation recursion, except for replacing the integrals with maximizations.

2.7 State Space Models

A state space model provides a general framework for analyzing many continuous-valued dynamical phenomena. The model consists of an underlying *state* $\boldsymbol{x}_t \in \mathbb{R}^n$ driven by a set of deterministic *control inputs* $\boldsymbol{u}_t \in \mathbb{R}^m$. The latent process produces a set of *observations* $\boldsymbol{y}_t \in \mathbb{R}^d$. A stochastic state space model additionally incorporates mutually independent and white process noise and measurement noise terms $\boldsymbol{e}_t$ and $\boldsymbol{w}_t$, respectively. We assume these noise processes are zero-mean with covariances Σ_t and R_t, respectively. The process noise term can be used to account for disturbances or uncertainties in the assumed dynamical model, while the measurement noise term models noisy observation mechanisms.

2.7.1 Standard Discrete-Time Linear-Gaussian State Space Formulation

A discrete-time *linear time-invariant* (LTI) state space model is given by:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{e}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t + \boldsymbol{w}_t. \end{aligned} \tag{2.140}$$

The time invariance of the model describes the fact that the parameters $\{A, B, C, D\}$ defining the linear state space model do not depend on the time index t. The terms $\boldsymbol{e}_t$ and $\boldsymbol{w}_t$ are noise processes which satisfy:

$$E\left[\begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{e}_i \\ \boldsymbol{w}_i \end{pmatrix}\begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{e}_j \\ \boldsymbol{w}_j \end{pmatrix}^T\right] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & \Sigma\delta_{ij} & S\delta_{ij} \\ 0 & S^T\delta_{ij} & R\delta_{ij} \end{bmatrix}. \tag{2.141}$$

This formulation ensures that the state sequence $\boldsymbol{x}_{1:T}$ forms a continuous-valued, discrete-time *wide-sense* Markov process [89]. Note, however, that $\boldsymbol{y}_{1:T}$ is *not* marginally wide-sense Markov although the joint process $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is. When $\boldsymbol{e}_t$ and $\boldsymbol{w}_t$ are Gaussian noise processes, implying that the second order statistics fully characterize the stochastic process, the state sequence forms a *strict-sense* Markov process: the state $\boldsymbol{x}_t$ yields the past, $\boldsymbol{x}_{1:t-1}$, and the future, $\boldsymbol{x}_{t+1:T}$, conditionally independent. Neither time-invariance nor linear dynamics is necessary for the strict-sense, discrete-time Markov process result.

In this thesis, we typically assume an uncontrolled (i.e., $\boldsymbol{u}_t = 0$) model:

$$\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + \boldsymbol{e}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \boldsymbol{w}_t. \end{aligned} \tag{2.142}$$

The graphical model for this process is equivalent to that of the hidden Markov model depicted in Fig. 2.8.

■ 2.7.2 Vector Autoregressive Processes

Many dynamical processes can be modeled as *autoregressive* (AR). That is, the observations are a noisy linear combination of some finite set of past observations plus additive white noise. An order r *vector* AR process, denoted by VAR(r), with observations $\boldsymbol{y}_t \in \mathbb{R}^d$, can be defined as

$$\boldsymbol{y}_t = \sum_{i=1}^{r} A_i \boldsymbol{y}_{t-i} + \boldsymbol{e}_t \qquad \boldsymbol{e}_t \sim \mathcal{N}(0, \Sigma). \tag{2.143}$$

Here, the observations depend linearly on the previous r observation vectors. We refer to $\{A_1, \dots, A_r\}$ as the set of *lag matrices*. Every VAR(r) process can be described in state space form by, for example, the following transformation:

$$\boldsymbol{x}_t = \begin{bmatrix} A_1 & A_2 & \dots & A_r \\ I & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \boldsymbol{e}_t \qquad \boldsymbol{y}_t = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t. \tag{2.144}$$

Note that there are many such equivalent *minimal* state space representations that result in the same input-output relationship [34, 110], where minimality implies that there does not exist a realization with lower state dimension (see Sec. 2.7.4 for further details). On the other hand, not every state space model may be expressed as a VAR(r) process for finite r [6]. We can thus conclude that considering a class of state space models with state dimension $r \cdot d$ and arbitrary dynamic matrix A subsumes the class of VAR(r) processes.

■ 2.7.3 Switching Linear Dynamic Systems

Many complex dynamical phenomena cannot be adequately described by a single linear dynamical model. However, the dynamics can often be approximated as switches between a set of linear systems in some probabilistic fashion based on an underlying, discrete-valued *mode* of the system. This class of *hybrid systems* is commonly referred to as a *jump-linear system*. When one takes the latent mode of the system to be a discrete-time Markov process, this model is typically referred to as a *Markov jump-linear system*

(MJLS) [30] or *switching linear dynamic system* (SLDS). Switched affine and piecewise affine (PWA) models, which we do not consider in this thesis, alternatively take the mode to be a function of the continuous state [130].

The SLDS we consider in this thesis can be described by:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ \boldsymbol{x}_t &= A^{(z_t)}\boldsymbol{x}_{t-1} + \boldsymbol{e}_t(z_t) \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \boldsymbol{w}_t, \end{aligned} \tag{2.145}$$

where z_t represents the mode of the system at time t, and is defined by a discrete-valued Markov process with transition distributions π_j. Here, we assume the process noise is mode-specific:

$$\boldsymbol{e}_t(z_t) \sim \mathcal{N}(0, \Sigma^{(z_t)}) \tag{2.146}$$

while the measurement mechanism is not. This assumption could be modified to allow for both a mode-specific measurement matrix $C^{(z_t)}$ and noise $\boldsymbol{w}_t(z_t) \sim \mathcal{N}(0, R^{(z_t)})$. However, such a choice is not always necessary nor appropriate for certain applications and can have implications on the identifiability of the model, as is discussed in Chapter 4.

We similarly define a *switching* VAR(r) process by

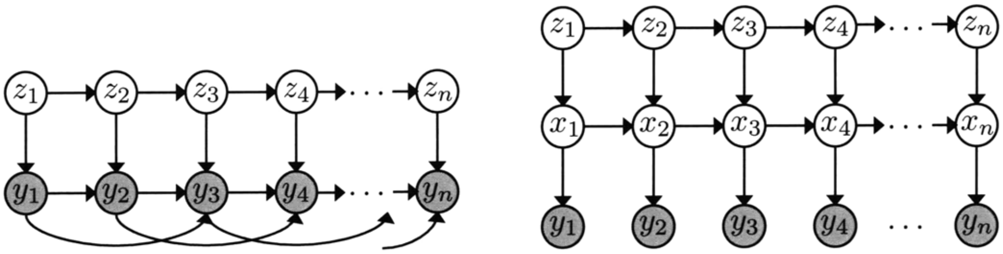
$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ \boldsymbol{y}_t &= \sum_{i=1}^{r} A_i^{(z_t)}\boldsymbol{y}_{t-i} + \boldsymbol{e}_t(z_t). \end{aligned} \tag{2.147}$$

Note that the underlying state dynamics of the SLDS are equivalent to a switching VAR(1) process.

Both the SLDS and the switching VAR process can be viewed as extensions of the standard HMM where instead of having conditionally independent observations given the mode sequence, the system has conditionally linear dynamics. See Fig. 2.10 for graphical model representations, and compare to that of the HMM in Fig. 2.8.

■ 2.7.4 Stochastic Realization Theory

The models we have described so far assume that the dynamic parameters are known and specified. The field of *stochastic realization theory* addresses the issue of constructing a model that *realizes* a stochastic process with a given set of statistical properties. One example of this is determining from a set of zero-mean, wide-sense stationary observations whether there exists an LTI state space model driven by white noise that produces the same second order moments, and if so, finding such a model. A question then arises as to the dimension of the underlying state of such a model, and more specifically, finding the minimal such dimension. The theory is developed assuming that the statistics of the process are available. In practice, however, applying the ideas



(a) (b)

Figure 2.10. Graphical models for the (a) switching vector autoregressive (VAR) process and (b) switching linear dynamical system (SLDS) over n time steps. For both models, a discrete-valued Markov process z_t dictates the linear dynamical model at time t. For an order r switching VAR process (shown here for $r = 2$), the conditionally linear dynamics are completely determined by the previous r observations. The SLDS instead relies on a latent, continuous-valued Markov state x_t to capture the history of the dynamical process.

of stochastic realization relies on approximations based on estimates of a finite set of these statistics (e.g., correlations over a fixed number of lags) from a finite set of data. Such practices fall under the category of *system identification*. *Model order reduction*, on the other hand, addresses the problem of finding a lower order approximation to a given state space model. In the following, we review some of the methodologies relevant to the models we consider. A more detailed presentation of this material can be found in [107]. We restrict our attention here to linear dynamical systems, with a brief overview of stochastic realization for the SLDS presented in Chapter 4.

Assume we are given a set of random variables $\boldsymbol{y}_{-\infty:\infty}$. Let

$$Y_+ = \begin{bmatrix} \boldsymbol{y}_t & \boldsymbol{y}_{t+1} & \boldsymbol{y}_{t+2} & \cdots \end{bmatrix} \tag{2.148}$$

$$Y_+ = \begin{bmatrix} \boldsymbol{y}_{t-1} & \boldsymbol{y}_{t-2} & \boldsymbol{y}_{t-3} & \cdots \end{bmatrix}. \tag{2.149}$$

Eq. (2.142) implies that the state of any stochastic realization of this process must yield the past and present conditionally independent:

$$p(\boldsymbol{y}_{-\infty,\infty} \mid \boldsymbol{x}_t) = p(Y_- \mid \boldsymbol{x}_t)p(Y_+ \mid \boldsymbol{x}_t). \tag{2.150}$$

Exploiting this Markov property, it can be shown that the size of the minimal state dimension that fully characterizes the second-order statistics of Y_+ and Y_- must be exactly the dimension of their cross-covariance:

$$\mathcal{H} \equiv \mathbb{E}[Y_+ Y_-^T] = \begin{bmatrix} \Lambda(1) & \Lambda(2) & \Lambda(3) & \cdots \\ \Lambda(2) & \Lambda(3) & \Lambda(4) & \cdots \\ \Lambda(3) & \Lambda(4) & \Lambda(5) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{2.151}$$

If we have a realization of the form given in Eq. (2.142), we see that

$$\Lambda(\tau) \equiv \mathbb{E}[\boldsymbol{y}_{t+\tau}\boldsymbol{y}_t^T] = \begin{cases} CP_xC^T + R, & \tau = 0; \\ CA^\tau P_xC^T \equiv CA^{\tau-1}G, & \tau > 0, \end{cases} \tag{2.152}$$

where P_x is the steady-state covariance satisfying (assuming A stable),

$$\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^T] \equiv P_x = AP_xA^T + \Sigma, \tag{2.153}$$

and we may rewrite the Hankel matrix as

$$\mathcal{H} = \begin{bmatrix} C \\ CA \\ \cdots \\ CA^{d-1} \end{bmatrix} \begin{bmatrix} G & AG & \cdots & A^{d-1}G \end{bmatrix} \equiv \mathcal{O}\mathcal{R}. \tag{2.154}$$

Noting that $\mathcal{O}$ and $\mathcal{R}$ correspond to extended observability and reachability matrices, respectively, for the system triplet (A, C, G), we may utilize the results from deterministic realization theory. These results inform us that a realization is minimal if and only if (A, C) is observable and (A, G) is reachable. Such a minimal realization is unique up to a change of basis in the coordinates of the state. Namely, we can define a set of equivalent minimal realizations as:

$$\mathcal{M}(A, C, G) = \{(CT, T^{-1}AT, T^{-1}G) \mid T \text{ invertible similarity matrix}\}. \tag{2.155}$$

For any such (A, C, G) to be a stochastic realization, it must also satisfy the following set of *positive real equations*:

$$P_x = AP_xA^T + \Sigma \succeq 0 \tag{2.156}$$

$$\Lambda(0) = CP_xC^T + R \succeq 0 \tag{2.157}$$

$$G = AP_xC^T. \tag{2.158}$$

The Kalman-Yakubovich-Popov (or positive real) lemma [91, 189] states that there exists a P_x satisfying the positive real equations if and only if the covariance $\Lambda(\tau)$ is a positive semidefinite function.

In practice, one does not have an infinite collection of correlations $\{\Lambda(\tau)\}_{\tau=1}^{\infty}$, but rather a finite set of observations $\boldsymbol{y}_{1:T}$ or correlations $\{\Lambda(\tau)\}_{\tau=1}^{T}$ from which one wants to produce a realization of the state space model. This necessarily leads to approximate methods since the above rank and factorization methods cannot be exactly applied to a finite Hankel matrix using the covariance estimates produced by $\boldsymbol{y}_{1:T}$. Instead, there is a wealth of literature that attempts to find a low-rank approximation to the finite Hankel matrix, such as through principle component analysis or more sophisticated balanced truncation techniques that aim to be coordinate-invariant. Note that such realizations need not satisfy the positive real equations and care must be taken to ensure that the state space realization is valid.

■ 2.7.5 Kalman Filtering and Smoothing

The *Kalman filter* [90] provides a recursive algorithm for estimating the underlying state of a linear-Gaussian state space model given a set of observations and fixed model parameters. Classically, the recursion is derived by exploiting the orthogonality principles that arise from preprocessing the observations with a whitening filter, and then building the linear least-squares estimate of the state from this white sequence. For a detailed derivation, and for numerous properties and extensions of the standard Kalman filter, see [89]. For the purposes of this background chapter, we simply present the algorithmic outline and then build the connection with a message passing formulation. Specifically, consider the state space model of Eq. (2.142). In Algorithm 2, we outline the standard Kalman filter for computing the linear least-squares estimate $\hat{\boldsymbol{x}}_{t|t}$ of $\boldsymbol{x}_t$ given $\boldsymbol{y}_1, \dots, \boldsymbol{y}_t$.

1. Initialize filter with

$$P_{0|0} = P_0$$
$$\hat{\boldsymbol{x}}_{0|0} = 0$$

2. Working forwards in time, for each $t \in \{1, \dots, T\}$:

   (a) Compute

$$K_t = P_{t|t-1}C(CP_{t|t-1}C + R)^{-1}$$

   (b) Predict

$$\hat{\boldsymbol{x}}_{t|t-1} = A\hat{\boldsymbol{x}}_{t-1|t-1}$$
$$P_{t|t-1} = AP_{t-1|t-1}A^T + \Sigma$$

   (c) Update

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + K_t(\boldsymbol{y}_t - C\hat{\boldsymbol{x}}_{t|t-1})$$
$$P_{t|t} = P_{t|t-1} - K_tCP_{t|t-1}$$

Algorithm 2. Kalman filter recursion for an LTI system.

Other inference tasks for which closed-form solutions exist include: *fixed-point smoothing* to form an estimator $\hat{\boldsymbol{x}}_{t^*|T}$ for t^* fixed and $T > t^*$; *fixed-lag smoothing* to form $\hat{\boldsymbol{x}}_{T-k|T}$ for k fixed; and *fixed-interval smoothing* to form $\hat{\boldsymbol{x}}_{t|T}$ for T fixed and for *all* $t < T$. For each of these problems, there are multiple possible algorithmic solutions. See [89] for more details.

Relationship to Belief Propagation

As with the forward-backward algorithm, the Kalman filter and the Rauch-Tung-Striebel variant of the Kalman smoother can be related to the belief propagation algorithm of Sec. 2.5.4. Similar derivations to the ones presented in this section will prove useful in the derivations in Appendices D and E.

For any state space model (i.e., model with a graphical representation as in Fig. 2.8), the following recursions exist:

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t) = \frac{p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_t \mid \boldsymbol{x}_t)}{p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_t)} \propto p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_{t-1}) \tag{2.159}$$

$$p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t) = \int_{\mathcal{X}_t} p(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t) d\boldsymbol{x}_t \tag{2.160}$$

$$\propto \int_{\mathcal{X}_t} p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t) p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t) d\boldsymbol{x}_t. \tag{2.161}$$

Assuming a linear-Gaussian state space model as in Eq. (2.142), these operations can be evaluated in closed form and simply correspond to operations on Gaussian mean and covariance parameters. Eq. (2.159) yields the *update* step of the Kalman filter, while Eq. (2.161), commonly referred to as the *Chapman-Kolmogorov equation*, yields the *predict* step. We will utilize these generic formulations in our derivation of the Kalman message passing algorithm.

We start by defining a forward message in a similar manner to that of the HMM forward-backward algorithm:

$$\alpha_{t+1}(\boldsymbol{x}_{t+1}) = \left[\int_{\mathcal{X}_t} p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t) \alpha_t(\boldsymbol{x}_t) d\boldsymbol{x}_t \right] \cdot p(\boldsymbol{y}_{t+1} \mid \boldsymbol{x}_{t+1}). \tag{2.162}$$

From Eq. (2.161), we can directly infer that, as with the HMM,

$$\alpha_t(\boldsymbol{x}_t) \propto p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t). \tag{2.163}$$

Assume $\alpha_t(\boldsymbol{x}_t) \sim \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^f_{t|t}, \Lambda^f_{t|t})$, where $\mathcal{N}^{-1}(\theta, \Lambda)$ denotes a Gaussian $\mathcal{N}(\mu, \Sigma)$ in information form with $\Lambda = \Sigma^{-1}$ and $\theta = \Sigma^{-1}\mu$. We may write the integrand of Eq. (2.162) in the following quadratic form:

$$p(\boldsymbol{x}_{t+1} | \boldsymbol{x}_t) \propto \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_{t+1} - A\boldsymbol{x}_t)^T \Sigma^{-1} (\boldsymbol{x}_{t+1} - A\boldsymbol{x}_t) \right\} \tag{2.164}$$

$$\propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} \Sigma^{-1} & -\Sigma^{-1}A \\ -A^T\Sigma^{-1} & A^T\Sigma^{-1}A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_t \end{bmatrix} \right\}$$

$$\alpha_t(\boldsymbol{x}_t) \propto \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_t - \theta^f_{t|t})^T \Lambda^f_{t|t} (\boldsymbol{x}_t - \theta^f_{t|t}) \right\} \tag{2.165}$$

$$\propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & \Lambda^f_{t|t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 \\ \theta^f_{t|t} \end{bmatrix} \right\}.$$

Combining these terms, the integrand is given by:

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\alpha_t(\boldsymbol{x}_t) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}_t - \theta^f_{t|t})^T\Lambda^f_{t|t}(\boldsymbol{x}_t - \theta^f_{t|t})\right\} \tag{2.166}$$

$$\propto \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{x}_{t+1}\\ \boldsymbol{x}_t\end{bmatrix}^T\begin{bmatrix}\Sigma^{-1} & -\Sigma^{-1}A\\ -A^T\Sigma^{-1} & A^T\Sigma^{-1}A+\Lambda^f_{t|t}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{t+1}\\ \boldsymbol{x}_t\end{bmatrix} + \begin{bmatrix}\boldsymbol{x}_{t+1}\\ \boldsymbol{x}_t\end{bmatrix}^T\begin{bmatrix}0\\ \theta^f_{t|t}\end{bmatrix}\right\}.$$

Marginalizing over $\boldsymbol{x}_{t+1}$ using the standard Gaussian marginalization identity

$$\int_{\mathcal{X}_2}\mathcal{N}^{-1}\left(\begin{bmatrix}x_1\\ x_2\end{bmatrix};\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix},\begin{bmatrix}\Lambda_{11} & \Lambda_{12}\\ \Lambda_{21} & \Lambda_{22}\end{bmatrix}\right)dx_2 = \mathcal{N}^{-1}(x_1;\theta_1-\Lambda_{12}\Lambda_{22}^{-1}\theta_2,\Lambda_{11}-\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}),$$

we obtain:

$$\int_{\mathcal{X}_t} p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)\alpha_t(\boldsymbol{x}_t)d\boldsymbol{x}_t \propto \mathcal{N}^{-1}(\boldsymbol{x}_{t+1};\theta_{t,t+1},\Lambda_{t,t+1}), \tag{2.167}$$

where

$$\begin{aligned}\theta_{t,t+1} &= \Sigma^{-1}A(A^T\Sigma^{-1}A+\Lambda^f_{t|t})^{-1}\theta^f_{t|t}\\ \Lambda_{t,t+1} &= \Sigma^{-1}-\Sigma^{-1}A(A^T\Sigma^{-1}A+\Lambda^f_{t|t})^{-1}A^T\Sigma^{-1}.\end{aligned} \tag{2.168}$$

We can write our likelihood term as:

$$p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{y}_{t+1}-C\boldsymbol{x}_{t+1})^TR^{-1}(\boldsymbol{y}_{t+1}-C\boldsymbol{x}_{t+1})\right\} \tag{2.169}$$

$$\propto \exp\left\{-\frac{1}{2}\boldsymbol{x}^T_{t+1}C^TR^{-1}C\boldsymbol{x}_{t+1}+\boldsymbol{x}^T_{t+1}C^TR^{-1}\boldsymbol{y}_{t+1}\right\} \tag{2.170}$$

To combine these terms, we simply add the information parameters:

$$\alpha_{t+1}(\boldsymbol{x}_{t+1}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{x}^T_t(\Lambda_{t,t+1}+C^TR^{-1}C)\boldsymbol{x}_t+\boldsymbol{x}^T_t(\theta_{t,t+1}+C^TR^{-1}\boldsymbol{y}_{t+1})\right\}. \tag{2.171}$$

Thus,

$$\begin{aligned}\theta^f_{t+1|t+1} &= \theta_{t,t+1}+C^TR^{-1}\boldsymbol{y}_{t+1}\\ &= \Sigma^{-1}A(A^T\Sigma^{-1}A+\Lambda^f_{t|t})^{-1}\theta^f_{t|t}+C^TR^{-1}\boldsymbol{y}_{t+1}\\ \Lambda^f_{t+1|t+1} &= \Lambda_{t,t+1}+C^TR^{-1}C\\ &= \Sigma^{-1}-\Sigma^{-1}A(A^T\Sigma^{-1}A+\Lambda^f_{t|t})^{-1}A^T\Sigma^{-1}+C^TR^{-1}C,\end{aligned} \tag{2.172}$$

```
1. Initialize filter with
       $\Lambda^f_{0|0} = P_0^{-1}$
       $\theta^f_{0|0} = 0$

2. Working forwards in time, for each $t \in \{1, \dots, T\}$:
   (a) Compute
       $M_t = A^{-T}\Lambda^f_{t|t}A^{-1}$
       $J_t = M_t(M_t + \Sigma^{-1})^{-1}$
       $L_t = I - J_t.$
   (b) Predict
       $\Lambda_{t-1,t} = L_{t-1}M_{t-1}L_{t-1}^T + J_{t-1}\Sigma^{-1}J_{t-1}^T$
       $\theta_{t-1,t} = L_{t-1}A^{-T}\theta^f_{t-1|t-1}$
   (c) Update
       $\Lambda^f_{t|t} = \Lambda_{t-1,t} + C^T R^{-1} C$
       $\theta^f_{t|t} = \theta_{t-1,t} + C^T R^{-1}\boldsymbol{y}_t$
```

Algorithm 3. Stable forward information form Kalman filter recursion.

which is equivalent to a standard update-to-update Kalman filter in information form with $\hat{\boldsymbol{x}}_{t|t} = (\Lambda^f_{t|t})^{-1}\theta^f_{t|t}$ and $P_{t|t} = (\Lambda^f_{t|t})^{-1}$. An equivalent form of this recursion (assuming A is invertible) is given by Algorithm 3.

We now examine the backward recursion. As in the HMM forward-backward algorithm, let

$$\beta_t(\boldsymbol{x}_t) = p(\boldsymbol{y}_{t+1}, \dots, \boldsymbol{y}_T \mid \boldsymbol{x}_t), \tag{2.173}$$

and recursively define

$$\beta_t(\boldsymbol{x}_t) \propto \int_{\mathcal{X}_{t+1}} p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t) p(\boldsymbol{y}_{t+1} \mid \boldsymbol{x}_{t+1}) \beta_{t+1}(\boldsymbol{x}_{t+1}) d\boldsymbol{x}_{t+1}. \tag{2.174}$$

Assuming $\beta_T(\boldsymbol{x}_t) \sim \mathcal{N}^{-1}(\boldsymbol{x}_T; \theta^b_{T|T+1}, \Lambda^f_{T|T+1})$, an analogous derivation to that of the forward recursion provides the following backwards recursion:

$$\begin{aligned} \theta^b_{t-1|t} &= A^T\Sigma^{-1}(\Sigma^{-1} + C^TR^{-1}C + \Lambda^b_{t|t+1})^{-1}(C^TR^{-1}\boldsymbol{y}_t + \theta^b_{t|t+1}) \\ \Lambda^b_{t-1|t} &= A^T\Sigma^{-1}A - A^T\Sigma^{-1}(\Sigma^{-1} + C^TR^{-1}C + \Lambda^b_{t|t+1})^{-1}\Sigma^{-1}A \end{aligned} \tag{2.175}$$

As in the HMM forward-backward algorithm, the posterior marginal is computed by combining the forward and backward messages, and then normalizing:

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \propto \alpha_t(\boldsymbol{x}_t)\beta_t(\boldsymbol{x}_t). \tag{2.176}$$

Replacing $\alpha_t(\boldsymbol{x}_t)$ and $\beta_t(\boldsymbol{x}_t)$ by their definitions in terms of the information parameters $\theta^f_{t|t}, \Lambda^f_{t|t}, \theta^b_{t|t+1}, \Lambda^b_{t|t+1}$, we have:

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \propto \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^f_{t|t}, \Lambda^f_{t|t})\mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^b_{t|t+1}, \Lambda^b_{t|t+1}) \tag{2.177}$$

$$\propto \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^f_{t|t} + \theta^b_{t|t+1}, \Lambda^f_{t|t} + \Lambda^b_{t|t+1}). \tag{2.178}$$

The connections between Kalman filtering and smoothing and belief propagation follow exactly as they did for the HMM. See Eq. (2.137), replacing z_n with $\boldsymbol{x}_t$ and using the definitions of $\alpha_t(\cdot)$ and $\beta_t(\cdot)$ above.

■ 2.8 Markov Chain Monte Carlo

As we have seen in Sec. 2.1, Bayesian inference (e.g., prediction or computation of posterior parameter estimates) relies on integration with respect to some potentially high-dimensional probability distribution[12]. We will generically denote this distribution by π. Except in the simplest cases, such integrals cannot be computed in closed form. *Markov chain Monte Carlo* (MCMC) methods [57, 142] provide a class of algorithms that produce estimates of the desired integral based on iterative sampling, combining *Monte Carlo* integration with samples from a specially constructed Markov chain. The key feature of these methods is that the sampling procedure does not rely on sampling from the distribution π, which is assumed to have an arbitrarily complex form.

■ 2.8.1 Monte Carlo Integration

The first step in understanding Monte Carlo integration involves formulating the desired integral as an expectation under the distribution π:

$$\int_{\mathcal{X}} f(x)\pi(x)dx = \mathbb{E}_\pi[f(x)]. \tag{2.179}$$

The Strong Law of Large Numbers [46] informs us that the sample average based on a set of independent samples $x_i \sim \pi$, $i = 1, \ldots, n$, converges almost surely to the

[12] Frequentists rely on integration for inference as well, and the techniques described in this section are equally well-suited to such problems. However, per the theme of this thesis, we will focus on the Bayesian framework.

true expectation under π. Thus, we may consider the following approximation, which becomes arbitrarily precise for n sufficiently large:

$$\mathbb{E}_{\pi}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i). \tag{2.180}$$

The assumption of having i.i.d. samples x_i can be relaxed by examination of *ergodic theory* [142]. The focus of MCMC methods is to develop an ergodic Markov chain with stationary distribution π, which we refer to as the *target* distribution, such that a sample path from this chain can be used to form the above estimate.

■ 2.8.2 The Metropolis-Hastings Algorithm

The *Metropolis-Hastings* algorithm provides a generic method for constructing an ergodic Markov chain, relying solely on defining a valid *proposal* distribution $q(\cdot \mid \cdot)$ and evaluation of the target distribution π up to a normalization constant. It is assumed evaluating $\pi(x)$ is easy, but sampling from this distribution is challenging. The weak conditions the proposal distribution must satisfy are described in Eq. (2.189)-Eq. (2.191) to follow. The Metropolis-Hastings algorithm is outlined in Algorithm 4.

Given a previous sample $x^{(t-1)}$:

1. Sample $x' \sim q(x' \mid x^{(t-1)})$.
2. Determine the acceptance probability:
$$\rho(x' \mid x^{(t-1)}) = \min\left\{ \frac{\pi(x')q(x^{(t-1)} \mid x')}{\pi(x^{(t-1)})q(x' \mid x^{(t-1)})}, 1 \right\}.$$
3. Sample
$$x^{(t)} \sim \rho(x' \mid x^{(t-1)})\delta_{x'} + \left(1 - \rho(x' \mid x^{(t-1)})\right)\delta_{x^{(t-1)}},$$
where δ_x is a Dirac mass at x.

Algorithm 4. Metropolis-Hastings algorithm.

The acceptance probability $\rho(y \mid x)$ is defined only when $\pi(x) > 0$. However, as long as $\pi(x^{(0)}) > 0$, the chain defined in Algorithm 4 will have $\pi(x^{(t)}) > 0$ for all t. We use the convention that $\rho(y \mid x)$ is 0 if both $\pi(x)$ and $\pi(y)$ are zero.

To analyze the properties of the Markov chain defined by the Metropolis-Hastings algorithm, it is useful to examine a condition known as detailed balance.

Proposition 2.8.1. *Let $\mathcal{K}(y \mid x) = p(x_{n+1} = y \mid x_n = x)$ be the* transition distribution *or* transition kernel *for a given Markov chain. If $\mathcal{K}(y \mid x)$ satisfies* detailed balance*:*

$$\mathcal{K}(y \mid x)\pi(x) = \mathcal{K}(x \mid y)\pi(y), \tag{2.181}$$

then the chain defined by this transition kernel has stationary distribution π. A Markov chain satisfying detailed balance is said to be reversible *with respect to π.*

Proof. Given a chain satisfying detailed balance,

$$\int \mathcal{K}(y \mid x)\pi(x)dx = \int \mathcal{K}(x \mid y)\pi(y)dx = \pi(y) \int \mathcal{K}(x \mid y)dx = \pi(y), \tag{2.182}$$

implying that π is indeed a stationary distribution of the Markov chain. ■

It is straightforward to show that the transition kernel defined by Algorithm 4 satisfies detailed balance. With probability $\rho(y \mid x)$, the chain transitions from x to a sample $y \sim q(y \mid x)$; otherwise, the chain transitions back to x. Thus, the kernel is a weighted mixture of the proposal distribution and a Dirac mass at x:

$$\mathcal{K}(y \mid x) = \rho(y \mid x)q(y \mid x) + \left(1 - \int \rho(z \mid x)q(z \mid x)dz\right)\delta_x. \tag{2.183}$$

To check the detailed balance condition of Eq. (2.181), we analyze each term of the transition kernel separately. The Dirac mass satisfies the following equality trivially:

$$\left(1 - \int \rho(z \mid x)q(z \mid x)dz\right)\delta_x\pi(x) = \left(1 - \int \rho(z \mid y)q(z \mid y)dz\right)\delta_y\pi(y). \tag{2.184}$$

We derive the equivalence of the other term in the resulting detailed balance equation as:

$$\begin{align}
\rho(y \mid x)q(y \mid x)\pi(x) &= \begin{cases} q(y \mid x)\pi(x), & q(y \mid x)\pi(x) < q(x \mid y)\pi(y); \\ q(x \mid y)\pi(y), & \text{otherwise.} \end{cases} \tag{2.185} \\
&= \min\left(q(y \mid x)\pi(x), q(x \mid y)\pi(y)\right) \tag{2.186} \\
&= \min\left(q(x \mid y)\pi(y), q(y \mid x)\pi(x)\right) \tag{2.187} \\
&= \rho(x \mid y)q(x \mid y)\pi(y). \tag{2.188}
\end{align}$$

Therefore, as long as

$$\bigcup_{x \in \text{ supp } \pi} \text{supp } q(\cdot \mid x) \supset \text{supp } \pi, \tag{2.189}$$

the chain defined by the Metropolis-Hastings algorithm (Algorithm 4) will satisfy detailed balance and thus define a Markov chain with π a stationary distribution. To prove that the Markov chain indeed converges to π (i.e., π is the unique invariant distribution for this chain and this distribution is reached from all initial states), we invoke some mild conditions under which the chain is both *aperiodic* and *Harris recurrent* [142]. Jointly, these conditions imply ergodicity.

A sufficient condition for the Metropolis-Hastings Markov chain to be aperiodic is for events $x^{(t)} = x^{(t-1)}$ to occur with some positive probability. That is,

$$P[\pi(x^{(t-1)})q(y \mid x^{(t-1)}) \leq \pi(y)q(x^{(t-1)} \mid y)] < 1. \tag{2.190}$$

Furthermore, if

$$q(y \mid x) > 0 \qquad \forall (x, y) \in \mathcal{X} \times \mathcal{X}, \tag{2.191}$$

then the Metropolis-Hastings Markov chain is *irreducible.* It can be shown (see Lemma 7.3 of [142]) that an irreducible Metropolis-Hastings chain is also Harris recurrent. Thus, any Metropolis-Hastings algorithm defined with a proposal distribution that satisfies the conditions of Eq. (2.189)-Eq. (2.191) will eventually produce samples from the stationary distribution π and

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} f(x^{(t)}) = \int_{\mathcal{X}} f(x)\pi(x)dx \qquad a.e. - \pi. \qquad . \tag{2.192}$$

Discussion on the rate of convergence to the stationary distribution can be found in [57, 142]. In general, this *burn-in* period is challenging to quantify, except by conservative bounds, and is especially challenging to assess in high-dimensional spaces. Convergence can be greatly affected by the initialization of the Markov chain, and in practice, it is common to run multiple chains from different initializations [50]. Multimodal target distributions with low valleys between the modes can lead to *poorly mixing* chains that stay in one region of the state space for long periods of time. Cleverly engineered proposal distributions, such as through tempering [125], can play a significant role in the success of a sampling algorithm.

■ 2.8.3 Gibbs Sampling

The *Gibbs sampler* [57, 142] is a special case of the Metropolis-Hastings algorithm in which the proposed sample is always accepted. The Gibbs sampler for n random variables $(x_1, x_2, \ldots, x_n)$ is summarized in Algorithm 5, from which we see that in order to sample from the full joint distribution on n random variables, it is sufficient to iteratively sample from each of the possibly univariate conditional distributions. As discussed in Sec. 2.5.2, a node in a directed graph is conditionally independent of all other nodes given its Markov blanket. Therefore, in the case of sparse graphs, the conditional density from which we are sampling is dependent only on a small subset of the other sampled nodes. We note that, as opposed to Metropolis-Hastings, the Gibbs sampler requires knowledge of the full conditional distributions and an ability to sample from them. Additionally, this algorithm is only applicable to models with at least two random variables.

```
Given a previous sample $\boldsymbol{x}^{(t-1)} = (x_1^{(t-1)}, \ldots, x_n^{(t-1)})$, generate:
  1. $x_1^{(t)} \sim p_1(x_1 \mid x_2^{(t-1)}, \ldots, x_n^{(t-1)})$.
  2. $x_2^{(t)} \sim p_2(x_2 \mid x_1^{(t)}, x_2^{(t-1)}, \ldots, x_n^{(t-1)})$.
                    $\vdots$
  n. $x_n^{(t)} \sim p_n(x_n \mid x_1^{(t)}, \ldots, x_{n-1}^{(t)})$.
```

Algorithm 5. Multi-stage Gibbs sampling algorithm.

To ensure a reversible chain[13], which leads to a Central Limit Theorem result for the estimator of Eq. (2.180) [142, 167], the *reversible Gibbs sampler* performs a sweep at every iteration from x_1 to x_n followed by a sweep in the reverse ordering back to x_1. Another variant on the standard Gibbs sampler of Algorithm 5, as proposed by Liu et al. [109], is to choose a random ordering for a single sweep—such an algorithm can lead to improved rates of convergence.

To make the connection between Gibbs sampling and Metropolis-Hastings, consider the proposal distribution at the i^{th} step of the sampler in Algorithm 5:

$$\begin{aligned} q_i(\boldsymbol{x}' \mid \boldsymbol{x}) = p_i(x_i' \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \\ \cdot \, \delta_{(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)}(x_1', \ldots, x_{i-1}', x_{i+1}', \ldots, x_n'), \end{aligned} \tag{2.193}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{x}' = (x_1', \ldots, x_n')$. That is, sample x_i' from its Markov kernel and set each x_j', $j \neq i$, equal to x_j. For this proposal, the acceptance probability is given by:

$$\begin{aligned} \rho_i(\boldsymbol{x}' \mid \boldsymbol{x}) &= \min\left\{ \frac{\pi(\boldsymbol{x}')q_i(\boldsymbol{x} \mid \boldsymbol{x}')}{\pi(\boldsymbol{x})q_i(\boldsymbol{x}' \mid \boldsymbol{x})}, 1 \right\} \\ &= \min\left\{ \frac{p_i(x_i' \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) p_i(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)}{p_i(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) p_i(x_i' \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)}, 1 \right\} \\ &= 1, \end{aligned}$$

implying that every proposed sample is accepted. Thus, one can interpret the full set of n steps of Algorithm 5 as a composition of n Metropolis-Hastings steps with Markovian kernels in which each proposal has acceptance probability equal to 1. If one were to treat all n steps of Algorithm 5 as a particular Metropolis-Hastings algorithm, the global acceptance probability of $\boldsymbol{x}' = (x_1', \ldots, x_n')$ is typically not equal to 1. However, with

[13] A non-reversible Markov chain implies that the chain does not satisfy the detailed balance condition of Eq. (2.181). However, since detailed balance is merely a sufficient condition for π to be a stationary distribution, this does not preclude π from being the stationary distribution of the Gibbs chain.

the direct form of Algorithm 5 in which each step is a Metropolis-Hastings proposal that is accepted (rather than modifying the algorithm to be a proposal distribution for a sample $\boldsymbol{x}'$), the convergence properties must be assessed differently than they were for the Metropolis-Hastings algorithm. In particular, each transition step does not satisfy the sufficient conditions of Eq. (2.190)-Eq. (2.191) so the resulting Markov chain is not necessarily irreducible. (Actually, each individual Markov kernel is *never* irreducible as it is constrained to a lower dimensional subspace.)

The following proposition states that, based on a condition of ergodicity, the Markov chain defined by Algorithm 5 indeed has stationary distribution π, as desired. We then provide a sufficient condition for ergodicity.

Proposition 2.8.2. *If $(\boldsymbol{x}^{(t)})$ is ergodic, then π is the stationary distribution of the chain defined in Algorithm 5.*

Proof. The kernel of the chain $(\boldsymbol{x}^{(t)})$ is

$$\mathcal{K}(\boldsymbol{x}' \mid \boldsymbol{x}) = p_1(x_1' \mid x_2, \ldots, x_n) p_2(x_2' \mid x_1', x_3, \ldots, x_n) \cdots p_n(x_n' \mid x_1', \ldots, x_{n-1}'). \quad (2.194)$$

Using this kernel, one can show that

$$\begin{aligned} P(\boldsymbol{x}' \in A) &= \int \mathbb{I}_A(\boldsymbol{x}') \mathcal{K}(\boldsymbol{x}' \mid \boldsymbol{x}) \pi(\boldsymbol{x}) d\boldsymbol{x}' d\boldsymbol{x} \\ &= \int_A \pi(\boldsymbol{x}') d\boldsymbol{x}', \end{aligned}$$

implying that π is the stationary distribution. See [142] for further details. ■

We now state a condition for the transition kernel defined by Algorithm 5, which, if satisfied, implies the ergodicity of the chain. Weaker conditions based on positivity constraints on the transition kernel exist (see Theorem 10.8 of [142]), but are more challenging to verify.

Proposition 2.8.3. *If the transition kernel associated with Algorithm 5 (see Eq.* (2.194)*) is absolutely continuous with respect to the dominating measure, the resulting chain is Harris recurrent.*

Proof. For a proof of this result, see [167]. ■

If one of the Gibbs steps is replaced with a Metropolis-Hastings step (i.e., a *hybrid* sampler), absolute continuity is lost and further analysis is necessary to conclude convergence of the resulting chain. Another important consideration is the fact that the developed Gibbs sampler does not apply to changing numbers of parameters since such changes imply irreducibility of the resulting chain. In such variable-dimension cases, one can instead appeal to *reversible jump* MCMC [60]. Both a hybrid sampler and reversible jump MCMC are employed in an algorithm developed in Chapter 5 (see Sec. 5.2 in particular.) However, the analysis of these techniques is beyond the scope of this background chapter, and we refer the interested reader to Chapters 10 and 11 of [142].

We conclude by noting that a two-stage Gibbs sampler ($n = 2$) has special convergence properties that do not apply in the general case of Algorithm 5. Some of these special properties arise from the fact that in the two-stage sampler, each subchain is also Markov allowing for component-wise study, which does not carry over to the more general case. For this two-stage sampler, instead of using the notation $x_1, \ldots, x_n$ as before, we use x and y to denote the two random variables of the model. An outline of the two-stage sampler is presented in Algorithm 6.

```
Given a previous sample $x^{(t-1)}$:
  1. Sample $y^{(t)} \sim p_{y|x}(y \mid x^{(t-1)})$.
  2. Sample $x^{(t)} \sim p_{x|y}(x \mid y^{(t)})$.
```

Algorithm 6. Two-stage Gibbs sampling algorithm.

From the construction of the sampler in Algorithm 6, it is clear that $(x^{(t)}, y^{(t)})$ forms a Markov chain. Interestingly, so does each subsequence $(x^{(t)})$ and $(y^{(t)})$, as previously suggested. The transition kernel for the sequence of random variables $(x^{(t)})$, for example, is

$$\mathcal{K}(x' \mid x) = \int p_{x|y}(x' \mid y) p_{y|x}(y \mid x) dy, \tag{2.195}$$

and the marginal distribution $p_x(\cdot)$ is indeed the stationary distribution of this chain:

$$\begin{aligned} p_x(x') &= \int p_{x|y}(x' \mid y) p_y(y) dy \\ &= \int p_{x|y}(x' \mid y) \int p_{y|x}(y \mid x) p_x(x) dx dy \\ &= \int \left[\int p_{x|y}(x' \mid y) p_{y|x}(y \mid x) dy \right] p_x(x) dx \\ &= \int \mathcal{K}(x' \mid x) p_x(x) dx. \end{aligned} \tag{2.196}$$

Based on such results, interleaving Markov chain results and the duality principle [36] apply. See Robert and Casella [142] for more details and for a full analysis of the convergence properties of the two-stage Gibbs sampler.

■ 2.8.4 Auxiliary, Blocked, and Collapsed Gibbs Samplers

In Sec. 2.8.3, we have assumed that it is feasible to sample from the full conditional distributions of the variables of interest, and we have assumed that this sampling has occurred by dividing the n random variables into n sampling stages. In this section, we explore several Gibbs sampling variants: *auxiliary*, *blocked*, and *collapsed*. In the auxiliary variable sampler, a set of *auxiliary variables*, which are not the random variables of interest in the inference, are added to the sampling procedure in order to enable closed-form sampling of the variables of interest. In some cases, one can improve the efficiencies of the sampler by *block sampling* multiple random variables jointly. Finally, *collapsed* Gibbs sampling involves the analytic marginalization of random variables from the model and then sampling the remaining variables from the reduced-order conditional distributions (assumed to maintain a simple, analytic form.) Each of these methods is summarized below.

Auxiliary Variable Sampling

There are some cases in which augmenting the random variables of interest x with *auxiliary variables* or *completion variables*[14] allows for closed form conditional distributions for the augmented set $y = \{x, z\}$. Note that although $(y^{(t)})$ forms a Markov chain (by construction), the subchain $(x^{(t)})$ need not. However, the subchain $(\boldsymbol{x}^{(t)})$ still converges to the the marginal distribution $p_{\mathsf{x}}(\cdot)$ (see Theorem 10.6 of [142].) In the standard mixture model we explore in Example 2.8.1, the completion variables z have a physical interpretation as the *mixture components* that allow for Gibbs sampling of the *mixture weights* $\{\pi_k\}$ and *mixture parameters* $\{\theta_k\}$. Estimation of the mixture weights and parameters can then be performed by simply discarding the completion variables.

Often, the auxiliary variables are only added for a subset of the sampling stages, and then discarded at other stages. That is, the auxiliary variables are sampled based on the current MCMC configuration of the variables of interest, then some subset of the variables of interest are sampled based on the sampled auxiliary variables. Finally, the auxiliary variables are discarded when sampling the other variables of interest that do not depend upon the auxiliary variables. Such sampling algorithms are developed in this thesis, for example, in Sec. 5.2 and Appendix C.1.

Blocked Gibbs Sampling

There are also many scenarios in which jointly sampling variables can lead to statistical efficiencies. Such *blocked Gibbs sampling* is especially useful when subsets of variables are very strongly dependent. In such cases, large moves in the joint probability space by standard coordinate-by-coordinate sampling may require stepping through deep valleys in the posterior distribution that can be avoided with blocked sampling of variables.

Collapsed Gibbs Sampling

Finally, in some cases it is possible to analytically marginalize *nuisance parameters* from the model and solely sample the variables of interest. In other cases, the variables

[14] The choice of terminology often depends upon whether the variables added have a physical interpretation in the model.

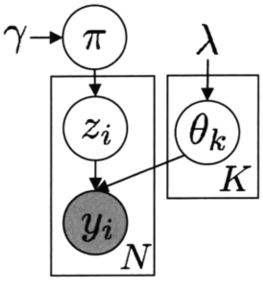


Figure 2.11. Graphical model of a finite mixture model in which the model parameters are defined with mixture weights $\boldsymbol{\pi} \mid \gamma \sim \mathrm{Dir}(\gamma/K, \dots, \gamma/K)$ and emission parameters $\theta_k \sim H, \lambda \sim H(\lambda)$. For each of the N observations y_i, a cluster assignment variable $z_i \in \{1, \dots, K\}$ is sampled as $z_i \mid \boldsymbol{\pi} \sim \boldsymbol{\pi}$, determining the mixture component for generating observations $y_i \mid \{\theta_k\}, z_i \sim F(\theta_{z_i})$.

marginalized are actually the variables of interest, and sampling occurs on a chain from which estimates of the desired variables can be formed. See Example 2.8.1. Analytical marginalization of variables in a Gibbs sampler, often referred to as *collapsed Gibbs sampling*, can aid in improving the mixing rate, especially when the marginalized random variables are high-dimensional as this can dramatically reduce the dimensionality of the search space.

However, there are scenarios in which such marginalization introduces dependencies between the remaining random variables that can actually lead to slower mixing rates. For example, let us consider the case of the hidden Markov model (HMM) described in Sec. 2.6. Given a sampled set of transition distributions π_j and emission parameters θ_j, one can employ a variant of the forward-backward message passing scheme to block-sample the entire state sequence $z_{1:T}$ (see Chapter 3.) On the other hand, if one chooses to marginalize the transition distributions (assuming a conjugate Dirichlet prior), then the state sequence no longer forms a simple Markov chain and thus block sampling is no longer feasible—one must instead rely on sequentially sampling the state z_t conditioned on the state at all other time steps $z_1, \dots, z_{t-1}, z_{t+1}, \dots, z_T$. Since the temporal correlations in an HMM can be quite strong, such sequential sampling can lead to very slow mixing rates. These mixing rate issues are examined in much further depth in Chapter 3, with a general empirical conclusion that block sampling of strongly dependent variables leads to more significant improvements in rates of convergence than marginalization.

Example 2.8.1. *Consider the finite mixture model of Fig. 2.11 in which each cluster assignment variable $z_i \in \{1, \dots, K\}$ indicates the mixture component associated with observations y_i. The model is defined by a set of mixture weights $\boldsymbol{\pi}$ distributed as*

$$\boldsymbol{\pi} \mid \gamma \sim Dir(\gamma/K, \dots, \gamma/K)$$

and emission parameters θ_k drawn as

$$\theta_k \mid H, \lambda \sim H(\lambda) \qquad k = 1, \dots, K.$$

Assume we have N observations. The generative model then dictates that each for each $i \in \{1, \ldots, N\}$, we draw:

$$z_i \mid \boldsymbol{\pi} \sim \boldsymbol{\pi}$$
$$y_i \mid \{\theta_k\}_{k=1}^{K}, z_i \sim F(\theta_{z_i}).$$

In what follows, we assume the distribution $F(\theta_k)$ has an associated conditional density $f(\cdot \mid \theta_k)$.

Let us assume that our goal is to infer the set of model parameters consisting of the mixture weights $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$ and the emission parameters $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^{K}$. One cannot simply employ a Gibbs sampler on these parameters since there do not exist closed-form conditional distributions $p(\boldsymbol{\pi} \mid \boldsymbol{\theta}, y_1, \ldots, y_N)$ and $p(\boldsymbol{\theta} \mid \boldsymbol{\pi}, y_1, \ldots, y_N)$. Instead, one could consider a completion *or* auxiliary variable *Gibbs sampler in which the cluster assignment variables $z_1, \ldots, z_N$ are additionally sampled.*

Let $z_{1:N} = \{z_1, \ldots, z_N\}$ and $z_{\setminus i} = \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_N\}$. Then, one can sample each z_i from

$$\begin{aligned} p(z_i = k \mid z_{\setminus i}, y_{1:N}, \boldsymbol{\pi}, \boldsymbol{\theta}) &= p(z_i = k \mid y_i, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ &\propto \pi_k f(y_i \mid \theta_k), \end{aligned}$$

with the first equality following from the Markov properties implied by the graphical model in Fig. 2.11. Conditioned on the cluster assignment variables $z_{1:N}$, the mixture weights $\boldsymbol{\pi}$ and emission parameters θ_k are mutually independent. The mixture weights can be sampled from the posterior Dirichlet distribution (see Eq. (2.74)*):*

$$p(\boldsymbol{\pi} \mid z_{1:N}, \gamma) = Dir(N_1 + \gamma/K, \ldots, N_K + \gamma/K) \qquad N_k = \sum_{i=1}^{N} \delta(z_i, k), \tag{2.197}$$

and the parameters θ_k from their associated posterior (depending upon the form of the prior $H(\lambda)$):

$$p(\theta_k \mid \{y_i \mid z_i = k\}, \lambda). \tag{2.198}$$

Here, we have used the fact that the full conditional distributions for $\boldsymbol{\pi}$ and each θ_k simply depend upon the sampled values of the random variables contained within the Markov blanket for that random variable's node. The resulting completion Gibbs sampler is outlined in Algorithm 7.

Alternatively, assuming the base measure $H(\lambda)$ is conjugate to the likelihood model F,[15] *one could analytically marginalize the mixture weights $\boldsymbol{\pi}$ and emission parameters $\boldsymbol{\theta}$ and solely sample the cluster assignment variables z_i. Based on a set of Gibbs samples of $z_{1:N}$, one could then estimate a set of model parameters (the variables of interest) from the distributions given in Eq.* (2.197) *and Eq.* (2.198)*.*

[15] Conjugacy of the prior on $\boldsymbol{\pi}$ to the multinomial observations z_i is already established by our choice of a Dirichlet prior.

Given mixture weights $\boldsymbol{\pi}^{(t-1)}$ and emission parameters $\{\theta_k^{(t-1)}\}_{k=1}^{K}$ from the previous Gibbs iteration, sample a new set of model parameters as follows:

1. For each $i \in \{1, \ldots, N\}$, independently assign observation y_i to one of the K clusters by sampling the cluster assignment variable z_i as:

$$z_i^{(t)} \sim \frac{1}{Z_i} \sum_{k=1}^{K} \pi_k^{(t-1)} f(y_i \mid \theta_k^{(t-1)}) \delta(z_i, k), \qquad Z_i = \sum_{k=1}^{K} \pi_k^{(t-1)} f(y_i \mid \theta_k^{(t-1)})$$

2. Sample a new set of mixture weights:

$$\boldsymbol{\pi}^{(t)} \sim \mathrm{Dir}(N_1 + \gamma/K, \ldots, N_K + \gamma/K), \qquad N_k = \sum_{i=1}^{N} \delta(z_i^{(t)}, k)$$

3. For each cluster $k \in \{1, \ldots, K\}$, independently sample new parameters from the conditional distribution implied by the observations currently assigned to that cluster:

$$\theta_k^{(t)} \sim p(\theta_k \mid \{y_i \mid z_i^{(t)} = k\}, \lambda)$$

Algorithm 7. Completion Gibbs sampler for the finite mixture model shown in Fig. 2.11. Each iteration resamples the cluster assignment variables for each of the N observations, and uses these sampled values to resample a set of mixture weights and emission parameters.

Integrating over $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, the Markov structure[16] *of the graph in Fig. 2.11 implies a posterior distribution on the cluster assignment variables that decomposes as:*

$$p(z_i \mid z_{\setminus i}, y_{1:N}, \gamma, \lambda) \propto p(z_i \mid z_{\setminus i}, \gamma) p(y_i \mid z_{1:N}, y_{\setminus i}, \lambda). \tag{2.199}$$

Based on the Dirichlet prior, the predictive distribution of Eq. (2.75) *informs us that:*

$$p(z_i = k \mid z_{\setminus i}, \gamma) = \frac{N_k^{-i} + \gamma/K}{N - 1 + \gamma} \qquad N_k^{-i} = \sum_{j \neq i} \delta(z_j, k). \tag{2.200}$$

When considering $z_i = k$, the likelihood term of Eq. (2.199) *simplifies to:*

$$p(y_i \mid z_i = k, z_{\setminus i}, y_{\setminus i}, \lambda) = p(y_i \mid \{y_j \mid z_j = k, j \neq i\}, \lambda). \tag{2.201}$$

Because $H(\lambda)$ is chosen conjugate to F, Eq. (2.201) *can be analytically determined. The resulting collapsed Gibbs sampler is outlined in Algorithm 8.*

[16] Marginalization of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ induces dependencies between the z_i and y_i not present in Fig. 2.11. However, the Markov structure of the graph in Fig. 2.11 can be exploiting during the integration over $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ to produce the decomposition of Eq. (2.199).

```
Given a previous set of cluster assignment variables $\boldsymbol{\pi}^{(t-1)}$, sequentially sample new assignments as follows:
1. Set $z_{1:N} = z_{1:N}^{(t-1)}$.
2. Sample a random permutation $\tau(\cdot)$ of the integers $\{1, \ldots, N\}$.
3. For each $i \in \{\tau(1), \ldots, \tau(N)\}$, resample $z_i$ as follows:
   (a) For each $k \in \{1, \ldots, K\}$, determine the predictive likelihood of observation $y_i$ based on an assignment to cluster $k$:
           $f_k(y_i) = p(y_i \mid \{y_j \mid z_j = k, j \neq i\}, \lambda)$.
       This likelihood can be computed from a set of cached sufficient statistics based on the results presented in Sec. 2.4.1.
   (b) Sample a new cluster assignment $z_i$ as:
           $z_i \sim \frac{1}{Z_i} \sum_{k=1}^{K} (N_k^{-i} + \gamma/K) f_k(y_i) \delta(z_i, k), \qquad Z_i = \sum_{k=1}^{K} (N_k^{-i} + \gamma/K) f_k(y_i),$
       where $N_k^{-i}$ is defined as in Eq. (2.200).
   (c) Update cached sufficient statistics to reflect the assignment of $y_i$ to cluster $z_i$.
4. Set $z_{1:T}^{(t)} = z_{1:T}$.
```

Algorithm 8. Collapsed Gibbs sampler for the finite mixture model shown in Fig. 2.11. Each iteration resamples the cluster assignment variables for each of the N observations, having analytically marginalized the mixture weights and emission parameters.

■ 2.9 Bayesian Nonparametric Methods

As motivated by the discussion of de Finetti's theorem (Theorem 2.1.2) in Sec. 2.1, it is theoretically desirable to consider models that are not limited to finite parameterizations, and in so doing one must define prior distributions on these infinite-dimensional objects. Bayesian nonparametric methods avoid the often restrictive assumptions of parametric models by defining distributions on function spaces such as that of probability measures. If suitably designed, these methods allow for efficient, data-driven posterior inference. For a review of Bayesian nonparametric inference, see [120, 157, 178]. In the following sections, we briefly describe some classes of Bayesian nonparametric methods: the Dirichlet process, its hierarchical extension, and the beta process.

■ 2.9.1 Dirichlet Processes

A Dirichlet process (DP) is a distribution on probability measures on a measurable space Θ. This stochastic process[17] is uniquely defined by a *base measure* H on Θ and a *concentration parameter* γ; we denote it by $\mathrm{DP}(\gamma, H)$. The Dirichlet process is formally defined by the distributions induced on finite partitions of Θ.

Theorem 2.9.1. *Let H be a probability distribution on a measurable space Θ, and γ a positive scalar. Consider a finite partition $\{A_1, \ldots, A_K\}$ of Θ:*

$$\bigcup_{k=1}^{K} A_k = \Theta \qquad A_j \cap A_k = \emptyset, \ j \neq k. \tag{2.202}$$

A random probability measure G_0 on Θ is a draw from a Dirichlet process if its measure on every finite partition follows a Dirichlet distribution:

$$(G_0(A_1), \ldots, G_0(A_K)) \mid \gamma, H \sim \mathit{Dir}(\gamma H(A_1), \ldots, \gamma H(A_K)). \tag{2.203}$$

For each such base measure H and concentration parameter γ, there exists a unique stochastic process satisfying the above conditions, which we denote by $\mathrm{DP}(\gamma, H)$.

Proof. The proof of the existence of the Dirichlet process was initially provided by Ferguson [41], who invoked Kolmogorov's consistency conditions to establish the existence of the Dirichlet process as a stochastic process with Dirichlet marginals. A more constructive definition of the Dirichlet process was given by Sethuraman [149]. ■

Using Eq. (2.72) along with Eq. (2.203), it is straightforward to establish that for any $A \subset \Theta$,

$$\mathbb{E}[G_0(A) \mid H] = H(A) \qquad G_0 \mid H, \gamma \sim \mathrm{DP}(\gamma, H)\,. \tag{2.204}$$

Based on an observation $\theta' \sim G_0$ that falls within an element A_k of a given partition $\{A_1, \ldots, A_K\}$, one can use the Dirichlet posterior analysis results of Eq. (2.74) to show that

$$(G_0(A_1), \ldots, G_0(A_K)) \mid \theta', H, \gamma \sim \mathrm{Dir}(\gamma H(A_1), \ldots, \gamma H(A_k) + 1, \ldots, \gamma H(A_K)). \tag{2.205}$$

Here, we note that the observation θ' only affects the Dirichlet parameter of the arbitrarily small partition element A_k in which it is contained. Formalizing such an analysis,

[17] In elementary probability theory, random variables are functions whose range is $\mathbb{R}$, whereas more advanced probability theory allows random variables to range over more general spaces (e.g., function spaces, spaces of probability measures, etc.). Stochastic process theory describes these more general random objects.

Ferguson [41] showed that a set of independent observations $\theta'_1, \dots, \theta'_N$, $\theta'_i \sim G_0$, leads to a posterior distribution

$$G_0 \mid \theta'_1, \dots, \theta'_N, H, \gamma \sim \mathrm{DP}\left(\gamma + N, \frac{\gamma}{\gamma + N} H + \frac{1}{\gamma + N} \sum_{i=1}^{N} \delta_{\theta'_i}\right), \tag{2.206}$$

where δ_θ denotes a unit-mass measure concentrated at θ. From Eq. (2.204), we see that for any $A \subset \Theta$,

$$\mathbb{E}[G_0(A) \mid \theta'_1, \dots, \theta'_N, H, \gamma] = \frac{\gamma}{\gamma + N} H(A) + \frac{1}{\gamma + N} \sum_{i=1}^{N} \delta_{\theta'_i}(A), \tag{2.207}$$

implying that,

$$\lim_{N \to \infty} \mathbb{E}[G_0(A) \mid \theta'_1, \dots, \theta'_N, H, \gamma] = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}(A), \tag{2.208}$$

where $\{\theta_k\}_{k=1}^{\infty}$ are the unique values in the set of observations $\{\theta'_i\}_{i=1}^{\infty}$, and β_k is the limiting empirical frequency of θ_k. Assuming the posterior concentrates about its mean, Eq. (2.208) implies that a realization from a Dirichlet process is discrete with probability one. Sethuraman [149] provides a formal proof of the discreteness of the Dirichlet process random measures G_0, and connects the weights β_k of this atomic measure with a constructive procedure.

Stick-Breaking Construction

Consider a probability mass function (pmf) $\{\beta_k\}_{k=1}^{\infty}$ on a countably infinite set, where the discrete probabilities are defined as follows:

$$\begin{aligned} v_k \mid \gamma &\sim \mathrm{Beta}(1, \gamma) & k &= 1, 2, \dots \\ \beta_k &= v_k \prod_{\ell=1}^{k-1} (1 - v_\ell) & k &= 1, 2, \dots. \end{aligned} \tag{2.209}$$

In effect, we have divided a unit-length stick into lengths given by the weights β_k: The k^{th} weight is a random proportion v_k of the remaining stick after the previous $(k-1)$ weights have been defined. This *stick-breaking construction* is generally denoted by $\beta \sim \mathrm{GEM}(\gamma)$. Sethuraman [149] showed that with probability one, a random draw $G_0 \sim DP(\gamma, H)$ can be expressed as

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \qquad \theta_k \mid H \sim H, \quad k = 1, 2, \dots, \tag{2.210}$$

From this definition, we see that the Dirichlet process actually defines a distribution over discrete probability measures. The stick-breaking construction also gives us insight into how the concentration parameter γ controls the relative proportion of the weights β_k.

Alternative stick-breaking processes have been studied for cases in which the weights v_k are drawn from a more general $\text{Beta}(a_k, b_k)$ distribution [72, 74]. When considering a two-parameter (a, b) family with $a_k = 1 - a$ and $b_k = b + ka$, one arrives at the Poisson-Dirichlet, or *Pitman-Yor*, process [137]. This process has heavier-tailed weight distributions than the Dirichlet process that have proven useful in applications such as natural language processing [58, 160], in which word frequencies closely follow a power-law.

Pólya Urn Predictions

The Dirichlet process has a number of properties which make inference based on this nonparametric prior computationally tractable. Once again, consider a set of observations $\{\theta'_i\}_{i=1}^N$

$$\theta'_i \mid G_0 \sim G_0. \tag{2.211}$$

Because probability measures drawn from a Dirichlet process are discrete, there is a strictly positive probability of multiple observations θ'_i taking identical values within the set $\{\theta_k\}_{k=1}^\infty$, with θ_k defined as in Eq. (2.210). Blackwell and MacQueen [18] introduced a Pólya urn representation of the θ'_i that results from integrating over the underlying random measure G_0 (distributed as in Eq. (2.204)):

$$\theta'_i \mid \theta'_1, \ldots, \theta'_{i-1} \sim \frac{\gamma}{\gamma + i - 1} H + \sum_{j=1}^{i-1} \frac{1}{\gamma + i - 1} \delta_{\theta'_j} \tag{2.212}$$

$$\sim \frac{\gamma}{\gamma + i - 1} H + \sum_{k=1}^{K} \frac{N_k}{\gamma + i - 1} \delta_{\theta_k}. \tag{2.213}$$

The second line is an equivalent representation of the first, but in terms of the unique set of parameter $\{\theta_k\}_{k=1}^\infty$, with N_k denoting the number of times each of these parameters was observed in the set $\{\theta'_i\}_{i=1}^N$.

A formal argument is presented in [18]. We can informally begin to justify Eq. (2.213) by once again considering Eq. (2.207). We first rewrite this expectation in terms of the unique parameters θ_k

$$\mathbb{E}[G_0(A) \mid \theta'_1, \ldots, \theta'_N, H, \gamma] = \frac{\gamma}{\gamma + N} H(A) + \frac{1}{\gamma + N} \sum_{k=1}^{\infty} N_k \delta_{\theta_k}(A). \tag{2.214}$$

Taking A to be the singleton set $\{\theta_k\}$, Eq. (2.214) implies that the marginal posterior probability of $\theta'_{N+1} = \theta_k$ for all k such that $N_k > 0$ is proportional to N_k, the number of times this parameter was previously observed. New parameter values are observed with probability proportional to γ.

The representation of Eq. (2.213) can be used to sample observations from a Dirichlet process without explicitly constructing the random probability measure $G_0 \sim \text{DP}(\gamma, H)$.

Chinese Restaurant Process

For each value θ'_i, let z_i be an indicator random variable that picks out the unique value θ_k such that

$$\theta'_i = \theta_{z_i}. \tag{2.215}$$

Eq. (2.213) implies the following predictive distribution on the indicator random variables:

$$p(z_{N+1} = z \mid z_1, \ldots, z_N, \gamma) = \frac{\gamma}{N+\gamma}\delta(z, K+1) + \frac{1}{N+\gamma}\sum_{k=1}^{K} N_k \delta(z,k), \tag{2.216}$$

where $N_k = \sum_{i=1}^{N} \delta(z_i, k)$ is the number of indicator random variables taking the value k, and $K+1$ is a previously unseen value.

The distribution on partitions induced by the sequence of conditional distributions in Eq. (2.216) is commonly referred to as the *Chinese restaurant process.* The analogy, which is useful in developing various generalizations of the Dirichlet process we consider in this thesis, is as follows. Take θ'_i to be a customer entering a restaurant with infinitely many tables, each serving a unique dish θ_k. Each arriving customer chooses a table, indicated by z_i, in proportion to how many customers are currently sitting at that table. With some positive probability proportional to γ, the customer starts a new, previously unoccupied table $K+1$. From the Chinese restaurant process, we see that the Dirichlet process has a reinforcement property that leads to a clustering at the values θ_k.

Number of Unique Observed Values

From Eq. (2.216) we see that when

$$z_i \mid \beta \sim \beta \qquad \beta \mid \gamma \sim \mathrm{GEM}(\gamma), \tag{2.217}$$

we can integrate out β to determine a closed-form predictive distribution for z_i. We can also find the distribution of the number of unique values of z_i (i.e., the number of occupied tables in the Chinese restaurant process) resulting from N draws from the measure β. Letting K denote the number of unique values of $\{z_1, \ldots, z_N\}$, Antoniak [5] derives this distribution to be:

$$p(K \mid N, \gamma) = \frac{\Gamma(\gamma)}{\Gamma(\gamma+N)} s(N,K)\gamma^K, \tag{2.218}$$

where $s(n,m)$ are unsigned Stirling numbers of the first kind [1].

Using Eq. (2.218), Antoniak [5] also observes that

$$\mathbb{E}[K \mid N, \gamma] \approx \gamma \log\left(\frac{\gamma+N}{\gamma}\right) \tag{2.219}$$

implying that the number of occupied tables in the Chinese restaurant process approaches (almost surely) $\gamma \log(N)$ as $N \to \infty$.

■ 2.9.2 Dirichlet Process Mixture Models

The Dirichlet process is commonly used as a prior on the parameters of a mixture model with a random number of components. Such a model is called a *Dirichlet process mixture model* and is depicted as a graphical model in Fig. 2.12(a)-(b). To generate observations, we choose

$$\begin{aligned} \theta'_i \mid G_0 &\sim G_0 \\ y_i \mid \theta'_i &\sim F(\theta'_i) \end{aligned} \tag{2.220}$$

for an indexed family of distributions $F(\cdot)$. This sampling process is also often described in terms of the indicator random variables z_i; in particular, we have

$$\begin{aligned} z_i \mid \beta &\sim \beta \\ y_i \mid \{\theta_k\}_{k=1}^{\infty}, z_i &\sim F(\theta_{z_i}). \end{aligned} \tag{2.221}$$

The parameter with which an observation is associated implicitly partitions or clusters the data. In addition, the Chinese restaurant process representation indicates that the Dirichlet process provides a prior that makes it more likely to associate an observation with a parameter to which other observations have already been associated. This reinforcement property is essential for inferring finite, compact mixture models. It can be shown under mild conditions that if the data were generated by a finite mixture, then the Dirichlet process posterior is guaranteed to converge (in distribution) to that finite set of mixture parameters [75]. See Sec. 6.2.5 for further discussion of the various asymptotic guarantees that have been established for models employing Dirichlet process priors.

Limit of Finite Mixture Models

We can also obtain the Dirichlet process mixture model as the limit of a sequence of finite mixture models, such as the one analyzed in Example 2.8.1. Let us assume that there are L components in a finite mixture model and we place a finite-dimensional, symmetric Dirichlet prior on these mixture weights:

$$\beta \mid \gamma \sim \text{Dir}(\gamma/L, \dots, \gamma/L). \tag{2.222}$$

Let $G_0^L = \sum_{k=1}^{L} \beta_k \delta_{\theta_k}$. Then, it can be shown [74, 76] that for every measurable function f integrable with respect to the measure H, this finite distribution G_0^L converges weakly to a countably infinite distribution G_0 distributed according to a Dirichlet *process.* That is,

$$\int_\theta f(\theta) dG_0^L(\theta) \xrightarrow{\mathcal{D}} \int_\theta f(\theta) dG_0(\theta), \tag{2.223}$$

as $L \to \infty$ for $G_0 \sim DP(\gamma, H)$. One can begin to justify this result by considering the K-component mixture model of Example 2.8.1 and taking the limit as $K \to \infty$ of

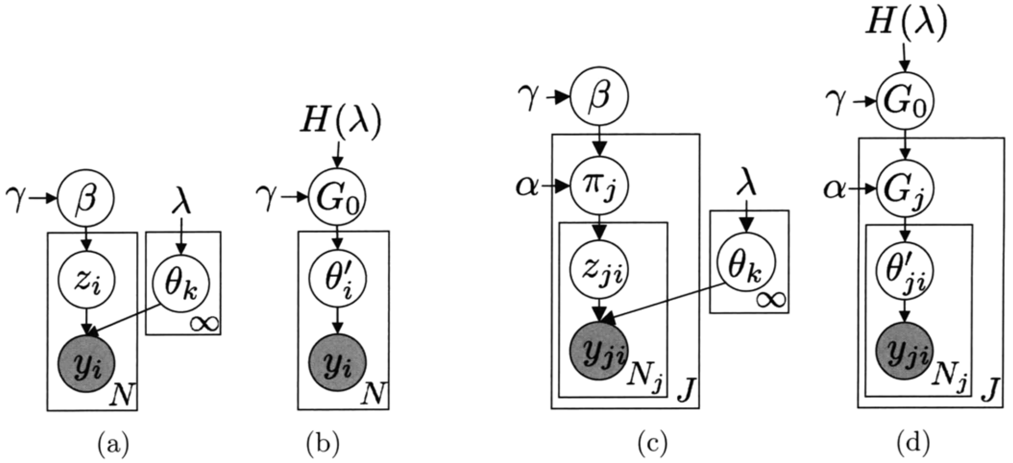


Figure 2.12. Dirichlet process (left) and hierarchical Dirichlet process (right) mixture models represented in two different ways as graphical models. (a) Indicator variable representation in which $\beta|\gamma \sim \text{GEM}(\gamma)$, $\theta_k|H,\lambda \sim H(\lambda)$, $z_i|\beta \sim \beta$, and $y_i|\{\theta_k\}_{k=1}^{\infty}, z_i \sim F(\theta_{z_i})$. (b) Alternative representation with $G_0|H,\gamma \sim \text{DP}(\gamma, H)$, $\theta'_i|G_0 \sim G_0$, and $y_i|\theta'_i \sim F(\theta'_i)$. (c) Indicator variable representation in which $\beta|\gamma \sim \text{GEM}(\gamma)$, $\pi_k|\alpha,\beta \sim \text{DP}(\alpha,\beta)$, $\theta_k|H,\lambda \sim H(\lambda)$, $z_{ji}|\pi_j \sim \pi_j$, and $y_{ji}|\{\theta_k\}_{k=1}^{\infty}, z_{ji} \sim F(\theta_{z_{ji}})$. (d) Alternative representation with $G_0|H,\gamma \sim \text{DP}(\gamma,H)$, $G_j|G_0 \sim \text{DP}(\alpha, G_0)$, $\theta'_{ji}|G_j \sim G_j$, and $y_{ji}|\theta'_{ji} \sim F(\theta'_{ji})$. The "plate" notation is used to compactly represent replication [162].

Eq. (2.200), resulting in:

$$p(z_i = k \mid z_{\setminus i}, \gamma) = \frac{N_k^{-i}}{N - 1 + \gamma} \tag{2.224}$$

for each instantiated cluster k. The probability of generating a new cluster is given the remaining mass $\gamma/(N-1+\gamma)$. Comparing these probabilities with those defined by Eq. (2.216) (using exchangeability to treat z_i as if it were the last observation), we see the equivalence of both predictive distributions.

In some scenarios, such as one we examine in Chapter 3, it is desirable to maintain a finite approximation to the Dirichlet process mixture model. One approach to producing such a finite approximation is simply to terminate the stick-breaking construction after some portion of the stick has already been broken and assign the remaining weight to a single component. This approximation is referred to as the *truncated Dirichlet process*. Another method, motivated by the convergence guarantee of Eq. (2.223), is to consider the *degree L weak limit approximation* to the Dirichlet process [76],

$$\text{GEM}_L(\alpha) \triangleq \text{Dir}(\alpha/L, \dots, \alpha/L), \tag{2.225}$$

where L is a number that exceeds the total number of expected mixture components. Both of these approximations, which are presented in [74, 76], encourage the learning of models with fewer than L components while allowing the generation of new components, upper bounded by L, as new data are observed. The two choices of approximations are compared in [102], and little to no practical differences are found.

■ 2.9.3 Hierarchical Dirichlet Processes

There are many scenarios in which groups of data are thought to be produced by related, yet distinct, generative processes. For example, take a sensor network monitoring an environment where time-varying conditions may influence the quality of the data. Data collected under certain conditions should be grouped and described by a similar, but different model from that of other data. The hierarchical Dirichlet process (HDP) [162] extends the Dirichlet process to such scenarios by taking a hierarchical Bayesian approach: the group-specific distributions G_j, with

$$G_j \mid G_0, \alpha \sim \mathrm{DP}(\alpha, G_0)\,, \tag{2.226}$$

are tied together via a global base measure G_0, which is itself given a Dirichlet process prior:

$$G_0 \mid H, \gamma \sim \mathrm{DP}(\gamma, H)\,, \tag{2.227}$$

As given by Eq. (2.204), for every $A \subset \Theta$,

$$\mathbb{E}[G_j(A) \mid G_0] = G_0(A). \tag{2.228}$$

In this sense, we can interpret G_0 as an "average" distribution across all groups. Below, we demonstrate that this specific choice of hierarchy implies that atoms are shared not only within groups, but also between groups, as desired. The HDP is depicted as a graphical model in Fig. 2.12(c)-(d).

Stick-Breaking Representation

Let $\{y_{j1}, \ldots, y_{jN_j}\}$ be the set of observations in group j. We assume there are J such groups of data. Then, replacing each Dirichlet process random measure with its associated stick-breaking representation (see Eq. (2.210)), the generative model can be written as:

$$\begin{array}{lll}
G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} & \beta \mid \gamma \sim \mathrm{GEM}(\gamma) & \\
 & \theta_k \mid H, \lambda \sim H(\lambda) & k = 1, 2, \ldots \\
 & & \\
G_j = \sum_{t=1}^{\infty} \tilde{\pi}_{jt} \delta_{\theta^*_{jt}} & \tilde{\pi}_j \mid \alpha \sim \mathrm{GEM}(\alpha) & j = 1, \ldots, J \\
 & \theta^*_{jt} \mid G_0 \sim G_0 & t = 1, 2, \ldots \\
 & & \\
\theta'_{ji} \mid G_j \sim G_j & y_{ji} \mid \theta'_{ji} \sim F(\theta'_{ji}) & j = 1, \ldots, J \\
 & & i = 1, \ldots, N_j.
\end{array} \tag{2.229}$$

See Fig. 2.12(d).

From this formulation, we clearly see how placing a Dirichlet process prior on G_0 creates a shared (and unbounded) support for each of the group-specific distributions

G_j. Namely, each group-specific set of support points θ^*_{jt} are drawn from the collection of atomic masses of G_0. Thus, there exists non-zero probability that different G_j share support points. If G_0 were instead absolutely continuous with respect to Lebesgue measure, there would be zero probability of the group-specific distributions having overlapping support.

Chinese Restaurant Franchise and the associated Table-Dish Representation

Teh et al. [162] have described the marginal probabilities obtained from integrating over the random measures G_0 and G_j. They show that these marginals can be described in terms of a *Chinese restaurant franchise* (CRF) that is an analog of the Chinese restaurant process. The CRF is comprised of J restaurants, each corresponding to an HDP group, and an infinite buffet line of dishes common to all restaurants. The process of seating customers at tables, however, is restaurant specific. To build up to this CRF, and to lay the foundation for modifications we make in Chapter 3, we first present the generative process in terms of indicator random variables being drawn from the stick-breaking measures β and $\tilde{\pi}_j$ leading to a *table-dish representation* of the HDP. We then marginalize these random measures to obtain the CRF.

More formally, we introduce indicator variables t_{ji} and k_{jt} to represent table and dish assignments. There are J restaurants (groups), each with infinitely many tables (clusters) at which customers (observations) sit. Each customer is pre-assigned to a given restaurant determined by that customer's group j. The table assignment for the i^{th} customer in the j restaurant is chosen as $t_{ji} \sim \tilde{\pi}_j$, and each table is assigned a dish (parameter) via $k_{jt} \sim \beta$. One can think of β as a set of ratings for the dishes served in the buffet line. Observation y_{ji} is then generated by global parameter

$$\theta'_{ji} = \theta^*_{jt_{ji}} = \theta_{k_{jt_{ji}}}. \tag{2.230}$$

The generative model for this table-dish representation is summarized below and is depicted as a graphical model in Fig. 2.13:

$$\begin{aligned} k_{jt} \mid \beta &\sim \beta \\ t_{ji} \mid \pi_j &\sim \tilde{\pi}_j \\ y_{ji} \mid \{\theta_k\}_{k=1}^{\infty}, \{k_{jt}\}_{t=1}^{\infty}, t_{ji} &\sim F(\theta_{k_{jt_{ji}}}). \end{aligned} \tag{2.231}$$

Marginalizing over the stick-breaking measures $\tilde{\pi}_j$ and β yields the following predictive distributions that describe the CRF:

$$\begin{aligned} p(t_{ji} \mid t_{j1}, \dots, t_{ji-1}, \alpha) &\propto \sum_{t=1}^{T_j} \tilde{n}_{jt}\delta(t_{ji}, t) + \alpha\delta(t_{ji}, T_j + 1) \\ p(k_{jt} \mid \underline{k}_1, \underline{k}_2, \dots, \underline{k}_{j-1}, k_{j1}, \dots, k_{jt-1}, \gamma) &\propto \sum_{k=1}^{K} m_{\cdot k}\delta(k_{jt}, k) + \gamma\delta(k_{jt}, K + 1), \end{aligned} \tag{2.232}$$

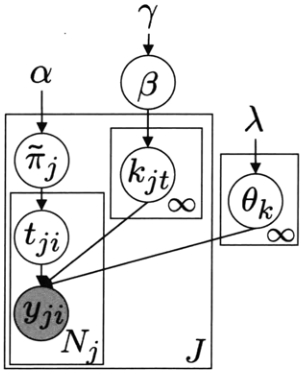


Figure 2.13. Graph of Chinese restaurant franchise (CRF). Customers y_{ji} sit at table $t_{ji}|\tilde{\pi}_j \sim \tilde{\pi}_j$. The first customer at each table chooses a dish $k_{jt}|\beta \sim \beta$.

where $m_{\cdot k} = \sum_j m_{jk}$ and $\underline{k}_j = \{k_{j1}, \ldots, k_{jT_j}\}$. Here, $\tilde{n}_{jt}$ denotes the number of customers in restaurant j sitting at table t, m_{jk} the number of tables in restaurant j serving dish k, T_j the number of currently occupied tables in restaurant j, and K the total number of unique dishes being served in the franchise. We note that $m_{\cdot k}$ is a pooling of the number of tables serving dish k in each of the individual restaurants, from which we see the sharing induced by the defined hierarchical model.

Eq. (2.232) implies that upon entering the j^{th} restaurant in the CRF, customer y_{ji} sits at currently occupied tables t_{ji} with probability proportional to the number of currently seated customers, or starts a new table $T_j + 1$ with probability proportional to α. Whenever a customer is the first customer to sit at a table in any of the J restaurants, that customer goes to the buffet line and picks a dish k_{jt} for their table, choosing the dish with probability proportional to the number of times that dish has been picked previously by any table in the franchise, or ordering a new dish θ_{K+1} with probability proportional to γ. The intuition behind this predictive distribution is that integrating over the dish ratings β results in customers making decisions based on the observed popularity of the dishes.

Compressed Indicator Variable Representation

Since each distribution G_j is drawn from a Dirichlet process with a discrete base measure G_0, multiple θ^*_{jt} may take an identical value θ_k for multiple unique values of t, implying that multiple tables in the same restaurant may be served the same dish, as depicted in Fig. 2.14. We can write G_j as a function of these unique dishes [162]:

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}, \quad \pi_j \mid \alpha, \beta \sim \mathrm{DP}(\alpha, \beta), \quad \theta_k \mid H \sim H, \tag{2.233}$$

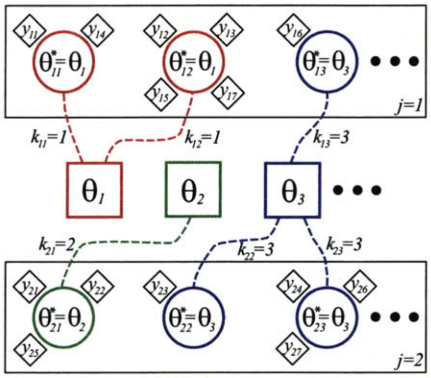


Figure 2.14. Chinese restaurant franchise (CRF) with $J = 2$ restaurants. The currently occupied tables each choose a dish $\theta^*_{jt}|G_j \sim G_j$, where $G_j|G_0, \alpha \sim \text{DP}(\alpha, G_0)$ is a discrete probability measure so that multiple tables may serve the same dish. Since G_1 has overlapping support with G_2, parameters (i.e., dishes) are shared between restaurants.

where π_j now defines a restaurant-specific distribution over dishes served rather than over tables, with

$$\pi_{jk} = \sum_{t|k_{jt}=k} \tilde{\pi}_{jt}. \tag{2.234}$$

Let z_{ji} be the indicator random variable for the unique dish eaten by customer y_{ji}, so that $z_{ji} = k_{jt_{ji}}$. A third equivalent representation of the generative model is in terms of these indicator random variables:

$$\begin{aligned} \pi_j \mid \alpha, \beta &\sim \text{DP}(\alpha, \beta) \\ z_{ji} \mid \pi_j &\sim \pi_j \\ y_{ji} \mid \{\theta_k\}, z_{ji} &\sim F(\theta_{z_{ji}}), \end{aligned} \tag{2.235}$$

and is shown in Fig. 2.12(c).

Limit of Finite Mixture Models

As with the Dirichlet process, the HDP mixture model has an interpretation as the limit of a finite mixture model. Placing a finite Dirichlet prior on β induces a finite Dirichlet prior on π_j (using Eq. (2.203) and the fact that $\pi_j \sim \text{DP}(\alpha, \beta)$):

$$\begin{aligned} \beta \mid \gamma &\sim \text{Dir}(\gamma/L, \ldots, \gamma/L) \\ \pi_j \mid \alpha, \beta &\sim \text{Dir}(\alpha\beta_1, \ldots, \alpha\beta_L). \end{aligned} \tag{2.236}$$

As $L \to \infty$, this model converges in distribution to the HDP mixture model [162].

■ 2.9.4 Beta Process

In Sec. 2.9.1, we described how the Dirichlet process, and its hierarchical extension, are useful in clustering applications (i.e., when it is assumed that the collection of observations are partitioned into a discrete set of classes, each described by a single parameter.) However, in many applications it is more appropriate to associate each observation with a binary *feature vector* indicating the *set* of parameters that describe the observation. For the clustering application, this vector would simply have a single 1 in the location corresponding to the index of the observation's cluster.

When given a large collection of observations, each described by multiple features, it is useful to consider a *featural model* that induces sparsity in the feature space by encouraging sharing of features among the observations. Analogous to the Dirichlet process inducing the Chinese restaurant process (CRP) clustering model with an unbounded number of clusters, we explain how a different stochastic process—the *beta process*—underlies the *Indian buffet process* (IBP) [62] featural model with an unbounded number of possible features[18]. Here, instead of associating a customer with a single dish as in the CRP, each customer of the IBP chooses a *collection* of dishes. And, just as the CRP encouraged the use of a sparse subset of the infinite collection of possible clusters, the IBP encourages the use of a sparse subset of the infinite feature space.

The Beta Process - Bernoulli Process Featural Model

The beta process [67, 161] is a stochastic process within the class of *completely random measures* [95, 96]; that is, evaluating a draw from a beta process over disjoint sets results in measures that are independent random variables. The definition of a completely random measure implies that the realizations are discrete, and thus described by a weighted collection of atoms, just as in the case of the Dirichlet process. We note, however, that Dirichlet process does *not* produce completely random measures since the weights of its realizations are constrained to sum to 1 (i.e., they are probability measures), inducing dependencies between the measures over disjoint sets. One can instead show [95] that Dirichlet process realizations are obtained by normalizing the completely random measures generated by the *gamma process.*

Consider a probability space Θ, and let B_0 denote a finite *base measure* on Θ with total mass $B_0(\Theta) = \alpha$. Supposing first that B_0 is absolutely continuous with respect to the dominating measure, we define the following *Lévy measure* [93, 106] on the product space $[0,1] \times \Theta$:

$$\nu(d\omega, d\theta) = c\omega^{-1}(1-\omega)^{c-1} d\omega B_0(d\theta) \tag{2.237}$$

Here, $c > 0$ is a *concentration parameter*; we denote such a beta process by $\mathrm{BP}(c, B_0)$.

[18] Specifically, the beta process is the de Finetti mixing distribution underlying the Indian buffet process (IBP), just as the Dirichlet process is the de Finitti distribution underlying the Chinese restaurant process. Historically, the IBP was introduced first by Griffiths and Ghahramani [62], who noted the exchangeability of the feature vectors. From Theorem 2.1.2, exchangeability implies there must exist an underlying measure yielding the feature vectors i.i.d.. As derived by Thibaux and Jordan [165], this measure is distributed according to the beta process.

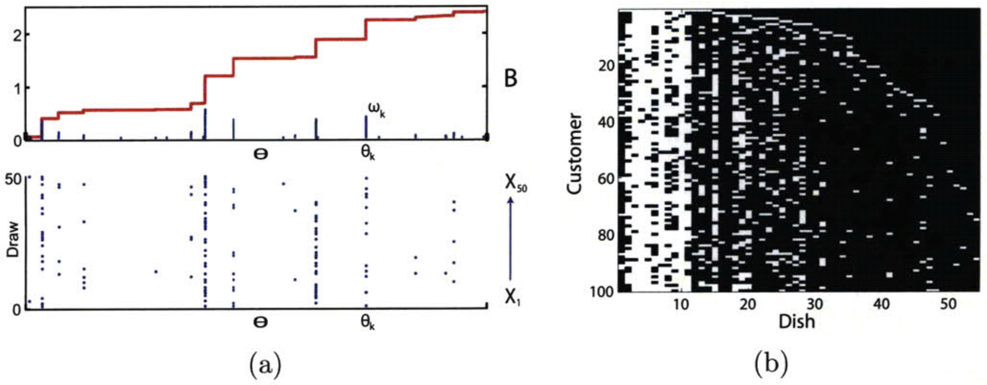


Figure 2.15. (a) *Top*: A draw B from a beta process is shown in blue, with the corresponding cumulative distribution in red. *Bottom*: 50 draws X_i from a Bernoulli process using the beta process realization. Each blue dot corresponds to a coin-flip at that atom in B that came up heads. (b) An image of a feature matrix associated with a realization from an Indian buffet process with $\alpha = 10$. Each row corresponding to a different customer, and each column a different dish. White indicates a chosen feature.

A draw $B \sim \mathrm{BP}(c, B_0)$ is then described by

$$B = \sum_{k=1}^{\infty} \omega_k \delta_{\theta_k}, \tag{2.238}$$

where $(\omega_1, \theta_1), (\omega_2, \theta_2), \dots$ are the set of atoms in a realization of a non-homogeneous Poisson process with rate measure ν. This set is necessarily infinite, as ν has infinite mass. However, because ν is σ-finite, Campbell's theorem [96] guarantees that for α finite, B has finite expected measure.

For a base measure B_0 containing atoms, the definition of the beta process measure B must be altered. Let $q_k \in (0,1)$ denote the mass of the k^{th} atom. A sample $B \sim \mathrm{BP}(c, B_0)$ necessarily contains this atom, with associated weight

$$\omega_k \sim \mathrm{Beta}(cq_k, c(1 - q_k)). \tag{2.239}$$

The overall realization B is then the sum of independent contributions from the continuous and discrete components of B_0. For an example realization and its associated cumulative distribution, see Fig. 2.15.

The beta process is conjugate to a class of so-called *Bernoulli processes* [165], denoted by $\mathrm{BeP}(B)$. A realization

$$X_i \mid B \sim \mathrm{BeP}(B), \tag{2.240}$$

with B an atomic measure (i.e., having a representation as in Eq. (2.238)), is a collection of unit mass atoms on Θ located at some subset of the atoms in B. In particular, for each atom θ_k in Eq. (2.238), we independently sample[19]

$$f_{ik} \sim \text{Bernoulli}(\omega_k) \tag{2.241}$$

and then set

$$X_i = \sum_k f_{ik}\delta_{\theta_k}. \tag{2.242}$$

Example realizations of $X_i \sim \text{BeP}(B)$, with B a draw from a beta process, are shown in Fig. 2.15(a).

For continuous measures B, we draw $L \sim \text{Poisson}(B(\Theta))$ and then independently sample a set of L atoms $\theta_\ell \sim B(\Theta)^{-1}B$. The Bernoulli realization is then given by:

$$X_i = \sum_{\ell=1}^{L} \delta_{\theta_\ell}. \tag{2.243}$$

In many applications, we interpret the atom locations θ_k as a shared set of global features. A Bernoulli process realization X_i then determines the subset of features allocated to object i:

$$\begin{aligned} B \mid B_0, c &\sim \text{BP}(c, B_0) \\ X_i \mid B &\sim \text{BeP}(B), \quad i = 1, \dots, N. \end{aligned} \tag{2.244}$$

Because beta process priors are conjugate to the Bernoulli process [165], the posterior distribution given N samples $X_i \sim \text{BeP}(B)$ is a beta process with updated parameters:

$$B \mid X_1, \dots, X_N, B_0, c \sim \text{BP}\left(c + N, \frac{c}{c+N}B_0 + \frac{1}{c+N}\sum_{i=1}^{N} X_i\right) \tag{2.245}$$

$$= \text{BP}\left(c + N, \frac{c}{c+N}B_0 + \sum_{k=1}^{K_+} \frac{m_k}{c+N}\delta_{\theta_k}\right) \tag{2.246}$$

Here, m_k denotes the number of objects X_i that select the k^{th} feature θ_k. For simplicity, we have reordered the feature indices to list first the K_+ features used by at least one object.

The Indian Buffet Process

Computationally, Bernoulli process realizations X_i are often summarized by an infinite vector of binary indicator variables $\boldsymbol{f}_i = [f_{i1}, f_{i2}, \dots]$, where $f_{ik} = 1$ if and only if object i exhibits feature k. As shown by Thibaux and Jordan [165], marginalizing over

[19] One can visualize this process as walking along the atoms of a discrete measure B and, at each atom θ_k, flipping a coin with probability of heads given by ω_k.

the beta process measure B, and taking $c = 1$, provides a predictive distribution on indicators equivalent to the Indian buffet process (IBP) of Griffiths and Ghahramani [62].

The IBP is a culinary analogy inspired by the Chinese restaurant process, which is itself the predictive distribution on partitions induced by the Dirichlet process [162]. The Indian buffet consists of an infinitely long buffet line of dishes, or features. The first arriving customer, or object, chooses Poisson(α) dishes. Each subsequent customer i selects a previously tasted dish k with probability m_k/i proportional to the number of previous customers m_k to sample it, and also samples Poisson(α/i) new dishes. The image of a feature matrix realization from an IBP with $\alpha = 10$ is shown in Fig. 2.15(b). Each row corresponding to a different customer, and each column a different dish. White indicates a chosen feature.

To derive the IBP from the beta process formulation described above, we note that the probability X_i contains feature θ_k after having observed $X_1, \dots, X_{i-1}$ is equal to the expected mass of that atom:

$$p(f_{ik} = 1 \mid X_1, \dots, X_{i-1}) = \mathbb{E}_{B|X_1,\dots,X_{i-1}}[p(f_{ik} = 1 \mid B)] = \mathbb{E}_{B|X_1,\dots,X_{i-1}}[\omega_k], \quad (2.247)$$

where our notation $\mathbb{E}_B[\cdot]$ means to take the expectation with respect to the distribution of B. Using the posterior distribution defined in Eq. (2.246), we consider the discrete and continuous portions of the base measure separately. The discrete component is a collection of atoms at locations $\theta_1, \dots, \theta_{K_+}$, each with weight

$$q_k = \frac{m_k}{c + i - 1}, \quad (2.248)$$

where K_+ is the number of unique atoms present in $X_1, \dots, X_{i-1}$. For each of the currently instantiated features $k \in \{1, \dots, K_+\}$, we have

$$\omega_k \sim \text{Beta}((c + i - 1)q_k, (c + i - 1)(1 - q_k)) \quad (2.249)$$

such that the expected weight is simply q_k, implying that the i^{th} object chooses one of the currently instantiated features with probability proportional to the number of objects that already chose that feature, m_k. We now consider the continuous portion of the base measure,

$$\frac{c}{c + i - 1} B_0. \quad (2.250)$$

The Poisson process defined by this rate function generates

$$\text{Poisson}\left(\frac{c}{c + i - 1} B_0(\Theta)\right) = \text{Poisson}\left(\frac{c}{c + i - 1}\alpha\right) \quad (2.251)$$

new atoms in X_i that do not appear in $X_1, \dots, X_{i-1}$. Following this argument, the first object simply chooses Poisson(α) features. If we specialize this process to $c = 1$, we arrive at the Indian buffet process of Griffiths and Ghahramani [62].

Just as with the Dirichlet process, hierarchical extensions [165] and stick-breaking constructions [163] of the Indian buffet process have been developed. However, we will not utilize such constructions in this thesis, so we omit the details of these processes.

Chapter 3

The Sticky HDP-HMM

HIDDEN Markov models (HMMs) have been a major success story in many applied fields; they provide core statistical inference procedures in areas as diverse as speech recognition, genomics, structural biology, machine translation, cryptanalysis and finance. Even after four decades of work on HMMs, however, significant problems remain. One lingering issue is the choice of the cardinality of the hidden state space. While standard parametric model selection methods can be adapted to the HMM, there is little understanding of the strengths and weaknesses of such methods in this setting, and practical applications of HMMs often fix the number of states using ad hoc approaches.

Recently, Teh et. al. [162] presented a Bayesian nonparametric approach to HMMs in which a stochastic process, the *hierarchical Dirichlet process* (HDP), defines a prior distribution on transition matrices over countably infinite state spaces. The resulting *HDP-HMM* is amenable to full Bayesian inference; in particular it is possible to compute and sample from posterior distributions over the number of model states. Moreover, this posterior distribution can be integrated over when making predictions, effectively averaging over models of varying complexity. The HDP-HMM has shown promise in a variety of applications, including visual scene recognition [97], music synthesis [68], and the modeling of genetic recombination [186] and gene expression [10].

One serious limitation of the standard HDP-HMM is that it inadequately models the temporal persistence of states. This problem arises in classical finite HMMs as well, where semi-Markovian models are often proposed as solutions. However, the problem is exacerbated in the nonparametric setting, in which the Bayesian bias towards simpler models is insufficient to prevent the HDP-HMM from giving high posterior probability to models with unrealistically rapid switching. As demonstrated in Fig. 3.1, HDP-HMM sampling algorithms often create redundant states and rapidly switch among them.

To illustrate the seriousness of this issue, let us consider a challenging application that we revisit in Sec. 3.4. The problem of *speaker diarization* involves segmenting an audio recording into time intervals associated with individual speakers [185]. This application seems like a natural fit for the HDP-HMM, as the number of true speakers is typically unknown, and may grow as more data are observed. However, this is not a setting in which model averaging is the goal; rather, it is critical to infer the number of speakers as well as the transitions among speakers. As we show in Sec. 3.4, the

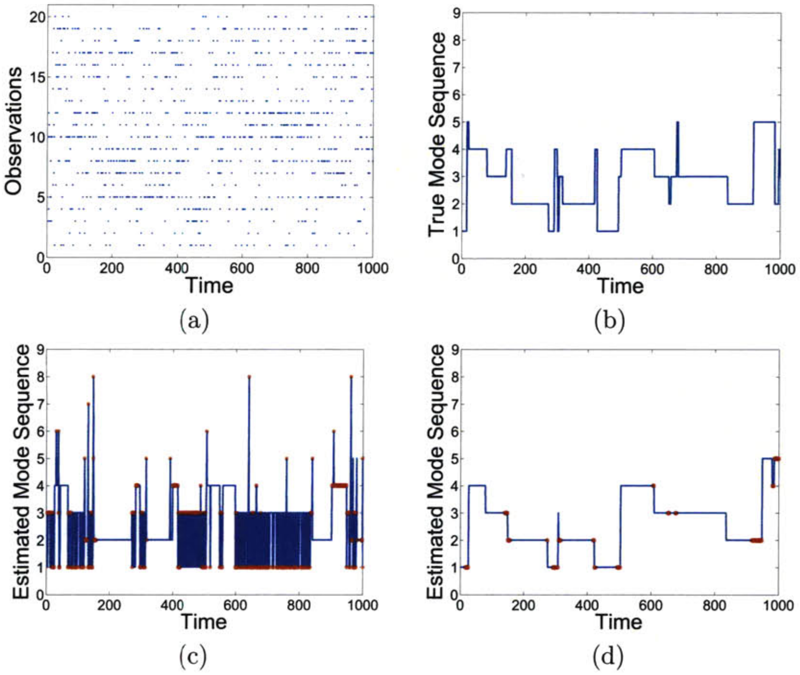


Figure 3.1. (a) Multinomial observation sequence; (b) true state sequence; (c)-(d) estimated state sequence after 30,000 Gibbs iterations for the original and sticky HDP-HMM, respectively, with errors indicated in red. Without an extra self–transition bias, the HDP-HMM rapidly transitions among redundant states.

HDP-HMM's tendency to rapidly switch among redundant states leads to poor speaker diarization performance.

In contrast, the methods we develop in Sec. 3.1 provide a general solution to the problem of state persistence in HDP-HMMs, and, along with other model extensions, yield a state-of-the-art speaker diarization method. The success on this challenging dataset is a profound demonstration of the efficacy of our methods for practical applications. The approach is easily stated—we simply augment the HDP-HMM to include a parameter for self-transition bias, and place a separate prior on this parameter. The challenge is to execute this idea coherently in a Bayesian nonparametric framework. Earlier papers have also proposed self-transition parameters for HMMs with infinite state spaces [11, 186], but did not formulate general solutions that integrate fully with Bayesian nonparametric inference.

Another goal of this chapter, which we explore in Sec. 3.3, is to develop a more fully nonparametric version of the HDP-HMM in which the emission distribution (the conditional distribution of observations given states) as well as the transition distribution is treated nonparametrically. This is again motivated by applications—classical applications of HMMs often find it necessary to use finite Gaussian mixtures as emission distributions in order to cope with multimodality. In the nonparametric setting it is natural to replace these finite mixtures with Dirichlet process mixtures (or with hierarchical Dirichlet process mixtures so as to tie emission distributions across states). Unfortunately, this idea is not viable in practice, because of the tendency of the HDP-HMM to rapidly switch between redundant states. By incorporating an additional self-transition bias, however, it is possible to make use of Dirichlet process mixtures for the emission distributions.

An important reason for the popularity of the classical HMM is its computational tractability. In particular, marginal probabilities and samples can be obtained from the HMM via an efficient dynamic programming algorithm known as the forward-backward algorithm (see Sec. 2.6.1). In Sec. 3.1.3 and Sec. 3.3.2, we show that this algorithm also plays an important role in computationally efficient inference for our generalized HDP-HMM. In particular, we develop a blocked Gibbs sampler which leverages forward–backward recursions to jointly resample the state and emission assignments for all observations.

■ 3.1 The HDP-HMM and Its Sticky Extension

Recall the hidden Markov model, or *HMM*, of Sec. 2.6. Once again, let z_t denote the state of the Markov chain at time t and π_j the state-specific transition distribution for state j. The HDP described in Sec. 2.9.3 can be used to develop an HMM with an infinite state space—the HDP-HMM [162]. Conceptually, we envision a doubly-infinite transition matrix, with each row corresponding to a Chinese restaurant of the metaphor introduced in Sec. 2.9.1. That is, the groups in the HDP formalism here correspond to states, and each Chinese restaurant defines a distribution on next states. The Chinese restaurant franchise (CRF) of Sec. 2.9.3 links these next-state distributions. Thus, in this application of the HDP, the group-specific distribution, π_j, is a state-specific transition distribution and, due to the infinite state space, there are infinitely many such groups. Since $z_t \sim \pi_{z_{t-1}}$, we see that z_{t-1} indexes the group to which y_t is assigned (i.e., all observations with $z_{t-1} = j$ are assigned to group j). Just as with the HMM, the current state z_t then indexes the parameter θ_{z_t} used to generate observation y_t. The generative model is summarized below, and is graphically depicted in Fig. 3.2(a).

$$\begin{aligned}
&\beta \mid \gamma \sim \mathrm{GEM}(\gamma) && \\
&\pi_j \mid \beta, \alpha \sim \mathrm{DP}(\alpha, \beta) && j = 1, 2, \ldots \\
&\theta_j \mid H, \lambda \sim H(\lambda) && j = 1, 2, \ldots \\
&z_t | \{\pi_j\}_{j=1}^{\infty}, z_{t-1} \sim \pi_{z_{t-1}} && t = 1, \ldots, T \\
&y_t | \{\theta_j\}_{j=1}^{\infty}, z_t \sim F(\theta_{z_t}) && t = 1, \ldots, T,
\end{aligned} \tag{3.1}$$

where we recall that GEM(·) denotes the stick-breaking construction (see Sec. 2.9.1).

By defining $\pi_j \sim \mathrm{DP}(\alpha, \beta)$, the HDP prior encourages states to have similar transition distributions. Namely, utilizing Eq. (2.204), the state-specific transition distribu-

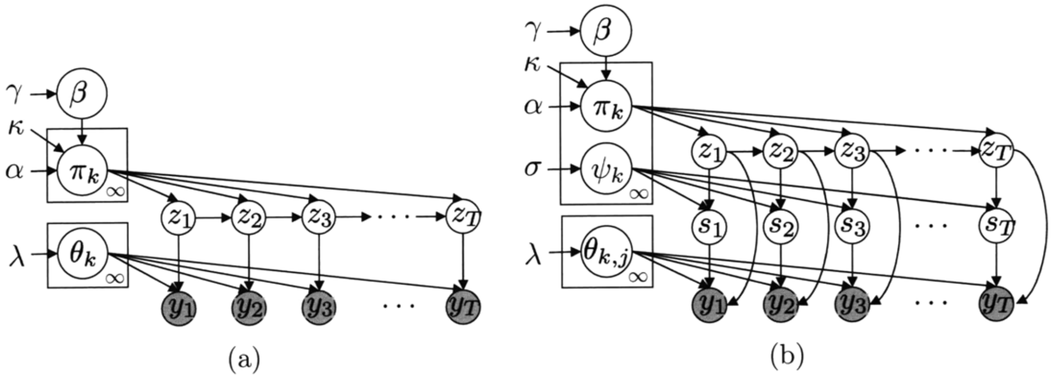


Figure 3.2. (a) Graphical representation of the sticky HDP-HMM. The state evolves as $z_{t+1} | \{\pi_k\}_{k=1}^{\infty}, z_t \sim \pi_{z_t}$, where $\pi_k | \alpha, \kappa, \beta \sim \mathrm{DP}(\alpha + \kappa, (\alpha\beta + \kappa\delta_k)/(\alpha + \kappa))$ and $\beta | \gamma \sim \mathrm{GEM}(\gamma)$, and observations are generated as $y_t | \{\theta_k\}_{k=1}^{\infty}, z_t \sim F(\theta_{z_t})$. The original HDP-HMM has $\kappa = 0$. (b) Sticky HDP-HMM with DP emissions, where s_t indexes the state-specific mixture component generating observation y_t. The DP prior dictates that $s_t | \{\psi_k\}_{k=1}^{\infty}, z_t \sim \psi_{z_t}$ for $\psi_k | \sigma \sim \mathrm{GEM}(\sigma)$. The j^{th} Gaussian component of the k^{th} mixture density is parameterized by $\theta_{k,j}$ so $y_t | \{\theta_{k,j}\}_{k,j=1}^{\infty}, z_t, s_t \sim F(\theta_{z_t,s_t})$.

tions are *identical* in expectation[1]:

$$\mathbb{E}[\pi_{jk} \mid \beta] = \beta_k. \tag{3.2}$$

Thus, the state-specific transition distributions π_j share sparsity in the state space, as induced by β. That is, the set of probable states visited from state i is related to that of state j. However, as we see from Eq. (3.2), the HDP-HMM does not differentiate self–transitions from moves between different states. When modeling data with state persistence, the flexible nature of the HDP-HMM prior allows for state sequences with unrealistically fast dynamics to have large posterior probability. For example, with multinomial emissions, a good explanation of the data is to divide different observation values into unique states and then rapidly switch between them (see Fig. 3.1). In such cases, many models with redundant states may have large posterior probability, thus impeding identification of a compact dynamical model which best explains the observations. The problem is compounded by the fact that once this alternating pattern has been instantiated by the sampler, its persistence is then reinforced by the properties of the Chinese restaurant franchise, thus slowing mixing rates. Furthermore, this fragmentation of data into redundant states can reduce predictive performance, as is discussed in Sec. 3.2. In many applications, one would like to be able to incorporate prior knowledge that slow, smoothly varying dynamics are more likely.

To address these issues, we propose to instead sample transition distributions π_j as

[1] In addition, the mean of these distributions, β, has, in expectation, a monotonically decreasing set of weights due to properties of its stick-breaking construction (see Sec. 2.9.1).

follows:

$$\begin{aligned}\beta \mid \gamma &\sim \mathrm{GEM}(\gamma)\\ \pi_j \mid \alpha, \kappa, \beta &\sim \mathrm{DP}\left(\alpha+\kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha+\kappa}\right).\end{aligned} \tag{3.3}$$

Here, $(\alpha\beta + \kappa\delta_j)$ indicates that an amount $\kappa > 0$ is added to the j^{th} component of $\alpha\beta$. Informally, what we are doing is increasing the expected probability of self-transition by an amount proportional to κ. Specifically, once again using Eq. (2.204), we see that the expected set of weights for transition distribution π_j is a convex combination of those defined by β and state-specific weight defined by κ:

$$\mathbb{E}[\pi_{jk} \mid \beta, \kappa] = \frac{\alpha}{\alpha+\kappa}\beta_k + \frac{\kappa}{\alpha+\kappa}\delta(j,k). \tag{3.4}$$

More formally, over a finite partition $(Z_1, \dots, Z_K)$ of the positive integers $\mathbb{Z}_+$, the definition of the Dirichlet process in Theorem 2.9.1 dictates that the prior on the measure π_j adds an amount κ only to the arbitrarily small partition containing j, corresponding to a self-transition. That is,

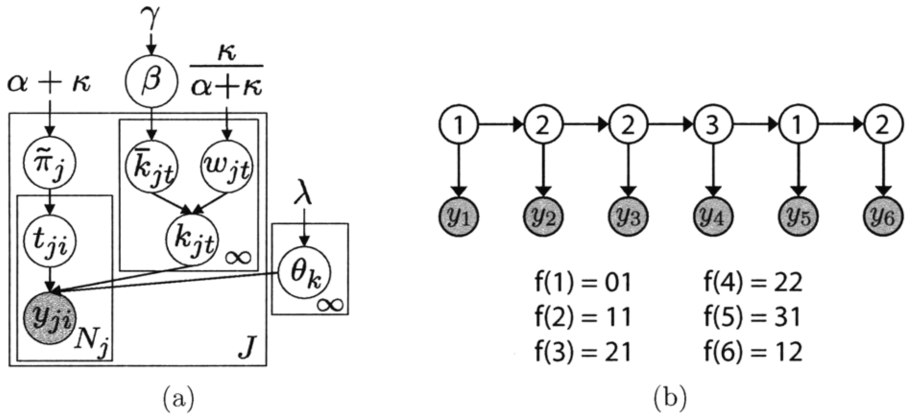
$$(\pi_j(Z_1), \dots, \pi_j(Z_K)) \mid \alpha, \beta \sim \mathrm{Dir}(\alpha\beta(Z_1) + \kappa\delta_j(Z_1), \dots, \alpha\beta(Z_K) + \kappa\delta_j(Z_K)) \tag{3.5}$$

When $\kappa = 0$ the original HDP-HMM of Teh et al. [162] is recovered. Because positive κ values increase the prior probability $\mathbb{E}[\pi_{jj} \mid \beta]$ of self–transitions, we refer to this extension as the *sticky* HDP-HMM. See Fig. 3.2(a).

The κ parameter is reminiscent of the self-transition bias parameter of the infinite HMM [11], a precursor of the HDP-HMM. The infinite HMM employs a two-level urn model. The top-level process places a probability on transitions to existing states in proportion to how many times these transitions have been seen, with an added bias towards a self-transition even if this has not previously occurred. With some remaining probability an oracle is called, representing the second-level urn. This oracle chooses an existing state in proportion to how many times the oracle previously chose that state, regardless of the state transition involved, or chooses a previously unvisited state. The oracle is included so that newly instantiated states may be visited from all currently instantiated states. While this urn model is an appealing description of probabilities on transitions, the lack of an underlying random measure makes it difficult to specify a coherent Bayesian inference procedure, and indeed the infinite HMM of Beal et al. [11] relies on a heuristic approximation to a Gibbs sampler. The full connection between HMMs on an infinite state space and an underlying Bayesian nonparametric prior, as well as the development of a coherent inference algorithm, was made in [162], but without the inclusion of a self-transition parameter (and hence with the potential pitfalls mentioned previously.)

■ 3.1.1 Chinese Restaurant Franchise with Loyal Customers

We extend the Chinese restaurant metaphor of Sec. 2.9.1 and Sec. 2.9.3 to the sticky HDP-HMM, where our franchise now has restaurants with loyal customers. In addition



(a) (b)

Figure 3.3. (a) Graph of CRF with loyal customers. Customers y_{ji} sit at table $t_{ji}|\tilde{\pi}_j \sim \tilde{\pi}_j$. Each table *considers* a dish $\bar{k}_{jt}|\beta \sim \beta$, but override variables $w_{jt}|\alpha, \kappa \sim \text{Ber}(\kappa/(\alpha+\kappa))$ can force the *served* dish k_{jt} to be j. (b) Mapping of time indices to CRF restaurant indices, with the state sequence labeled with a fixed set of assignments $z_{1:6} = [1, 2, 2, 3, 1, 2]$. For example, y_4 is seated in restaurant $j = 2$ (since $z_3 = 2$ implies $z_4 \sim \pi_2$), and is the second customer to be seated in that restaurant. As such, y_4 is assigned a restaurant index y_{22}. Observation y_1 is assigned to a special initial restaurant $j = 0$ due to the fact that z_1 is drawn from an initial distribution π^0.

to providing intuition for the predictive distribution on assignment variables, developing this metaphor aids in constructing the Gibbs samplers of Sec. 3.1.2 and Sec. 3.1.3. In the CRF with loyal customers, each restaurant in the franchise has a specialty dish with the same index as that of the restaurant. Although this dish is served elsewhere, it is more popular in the dish's namesake restaurant. We see this increased popularity in the specialty dish from the fact that a table's dish is now drawn from the *modified* dish ratings, namely,

$$k_{jt} \mid \alpha, \kappa, \beta \sim \frac{\alpha\beta + \kappa\delta_j}{\alpha+\kappa}. \tag{3.6}$$

Specifically, we note that each restaurant has a set of restaurant-specific ratings of the buffet line that redistributes the shared ratings β so that there is more weight on the house-specialty dish.

Recall that while customers in the CRF of the HDP are pre-partitioned into restaurants based on the fixed group assignments, in the HDP-HMM the value of the state z_{t-1} determines the group assignment (and thus restaurant) of customer y_t. Therefore, we will describe a generative process that first assigns customers to restaurants and then assigns customers to tables and dishes. We will refer to z_t as the parent and z_{t+1} as the child. The parent enters a restaurant j determined by its parent (the grandparent), $z_{t-1} = j$. We assume there is a bijective mapping $f : t \to ji$ of time indices t to restaurant/customer indices ji. See Fig. 3.3(b) for an example. The parent then chooses a table $t_{ji} \sim \tilde{\pi}_j$ and that table is served a dish indexed by k_{jt}. Noting that $z_t = z_{ji} = k_{jt_{ji}}$ (i.e., the value of the state is the dish index), the increased popularity of the house specialty dish implies that children are more likely to eat in the same restaurant as their parent and, in turn, more likely to eat the restaurant's specialty dish. This develops family loyalty to a given restaurant in the franchise. However, if the parent chooses a dish other than the house specialty, the child will then go to the restaurant where this dish is the specialty and will in turn be more likely to eat this dish, too. One might say that for the sticky HDP-HMM, children have similar tastebuds to their parents and will always go the restaurant that prepares their parent's dish best. Often, this keeps many generations eating in the same restaurant.

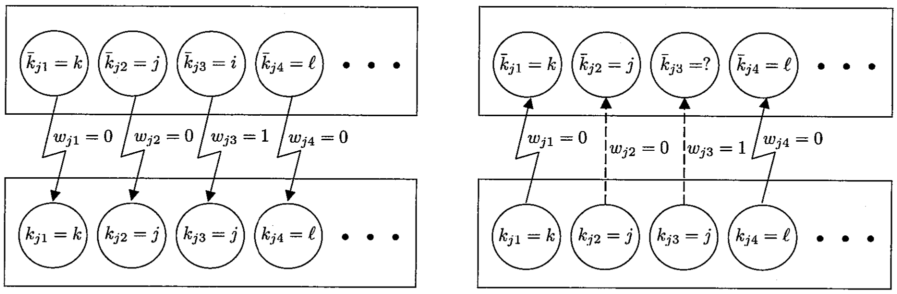
The inference algorithm for the sticky HDP-HMM, which is derived in Sec. 3.1.2, is simplified if we introduce a set of auxiliary random variables $\bar{k}_{jt}$ and w_{jt} as follows:

$$\begin{aligned} \bar{k}_{jt} \mid \beta &\sim \beta \\ w_{jt} \mid \alpha, \kappa &\sim \text{Ber}\left(\frac{\kappa}{\alpha+\kappa}\right) \triangleq \text{Ber}(\rho) \\ k_{jt} \mid \bar{k}_{jt}, w_{jt} &= \begin{cases} \bar{k}_{jt}, & w_{jt}=0; \\ j, & w_{jt}=1, \end{cases} \end{aligned} \tag{3.7}$$

where $\text{Ber}(\rho)$ represents the Bernoulli distribution with parameter ρ. Here, we have defined a self-transition parameter $\rho = \kappa/(\alpha+\kappa)$. The table first chooses a dish $\bar{k}_{jt}$ without taking the restaurant's specialty into consideration (i.e., the original CRF). With some probability, this *considered* dish is overridden (perhaps by a waiter's suggestion) and the table is served the specialty dish j. Thus, k_{jt} represents the *served* dish. We refer to w_{jt} as the *override* variable. For the original HDP-HMM, when $\kappa = 0$, the considered dish is always the served dish since $w_{jt} = 0$ for all tables. This generative process is depicted in Fig. 3.4(a). Our inference algorithm, described in Sec. 3.1.2, aims to infer these variables conditioned on knowledge of the *served* dishes k_{jt}. For example, if the served dish of table t in restaurant j is indexed by j, the house specialty, the origin of this dish may either have been from considering $\bar{k}_{jt} = j$ or having been overridden by $w_{jt} = 1$. See Fig. 3.4(b).

A table-dish representation of the sticky HDP-HMM, analogous to that of Fig. 2.13 for the HDP, is shown in Fig. 3.3(a). Although not explicitly represented in this graph, the sticky HDP-HMM still induces a Markov structure on the indicator random variables z_t, which, based on the value of the parent state z_{t-1}, are mapped to a group-specific index ji. This process is depicted in Fig. 3.3(b). One can derive a distribution on partitions by marginalizing over the stick-breaking distributed measures $\tilde{\pi}_j$ and β, just as in the HDP (see Eq. (2.232)). The CRF with loyal customers is then described by:

$$\begin{aligned} p(t_{ji} \mid t_{j1}, \dots, t_{ji-1}, \alpha, \kappa) &\propto \sum_{t=1}^{T_j} \tilde{n}_{jt} \delta(t_{ji}, t) + (\alpha+\kappa)\delta(t_{ji}, T_j+1) \\ p(\bar{k}_{jt} \mid \underline{\bar{k}}_1, \dots, \underline{\bar{k}}_{j-1}, \bar{k}_{j1}, \dots, \bar{k}_{jt-1}, \gamma) &\propto \sum_{k=1}^{\bar{K}} \bar{m}_{\cdot k} \delta(\bar{k}_{jt}, k) + \gamma \delta(\bar{k}_{jt}, \bar{K}+1), \end{aligned} \tag{3.8}$$



(a) Generative

(b) Inference

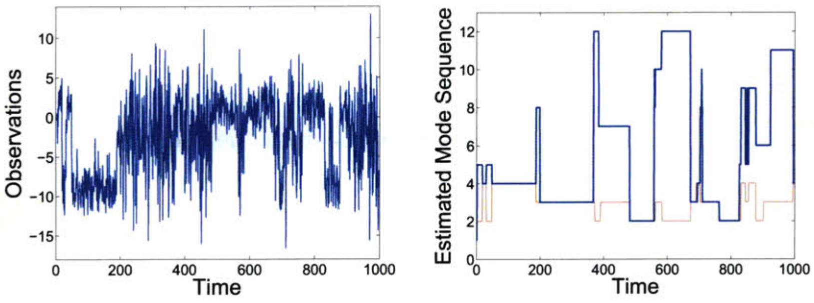
Figure 3.4. (a) Generative model of considered dish indices $\bar{k}_{jt}$ (top) being converted to served dish indices k_{jt} (bottom) via override variables w_{jt}. (b) Perspective from the point of view of an inference algorithm that must infer $\bar{k}_{jt}$ and w_{jt} given k_{jt}. If $k_{jt} \neq j$, then the override variable w_{jt} is automatically 0 implying that $\bar{k}_{jt} = k_{jt}$, as indicated by the jagged arrow. If instead $k_{jt} = j$, then this could have arisen from the considered dish $\bar{k}_{jt}$ being overridden ($w_{jt} = 1$) or not ($w_{jt} = 0$). These scenarios are indicated by the dashed arrow. If the considered dish was not overridden, then $\bar{k}_{jt} = k_{jt} = j$. Otherwise, $\bar{k}_{jt}$ could have taken any value, as denoted by the question mark.

where $\bar{m}_{jk}$ is the number of tables in restaurant j that *considered* dish k, and $\bar{K}$ the number of unique considered dishes in the franchise. The distributions on w_{jt} and k_{jt} remain as before, so that considered dishes are sometimes overridden by the house specialty.

■ 3.1.2 Sampling via Direct Assignments

In this section we present an inference algorithm for the sticky HDP-HMM of Sec. 3.1 and Fig. 3.2(a) that is a modified version of the direct assignment collapsed Gibbs sampler of [162]. This sampler circumvents the complicated bookkeeping of the CRF by sampling indicator random variables directly. That is, the sampler uses the condensed indicator variable representation of the HDP instead of the table-dish representation (see Sec. 2.9.3). The resulting sticky HDP-HMM direct assignment Gibbs sampler is outlined in Algorithm 9, with the full derivation presented in Appendix A.

The basic idea is that we marginalize over the infinite set of state-specific transition distributions π_k and parameters θ_k, and sequentially sample the state z_t given all other state assignments $z_{\setminus t}$, the observations $y_{1:T}$, and the global transition distribution β. A variant of the Chinese restaurant process gives us the prior probability of an assignment of z_t to a value k based on how many times we have seen other transitions from the previous state value z_{t-1} to k and k to the next state value z_{t+1}. We denote the number of transitions from j to k in $z_{1:T}$ by n_{jk}. As presented in Algorithm 9, this conditional distribution is dependent upon whether either or both of the transitions z_{t-1} to k and k to z_{t+1} correspond to a self-transition, most strongly when $\kappa > 0$. The prior probability of an assignment of z_t to state k is then weighted by the likelihood of the observation y_t given all other observations assigned to state k.



(a) (b)

Figure 3.5. Plots showing the sequential Gibbs sampler splitting temporally separated examples of the same true state into multiple estimated states. (a) Observation sequence. (b) Example of the estimated HMM state sequence (blue) at Gibbs iteration 1000 overlayed on the true HMM state sequence (red). Here, we see that a single true state is divided into multiple estimated states, each with high probability of self-transition.

Given a sample of the state sequence $z_{1:T}$, we can represent the posterior distribution of the global transition distribution β via a set of auxiliary random variables $\bar{m}_{jk}$, m_{jk}, and w_{jt}, which correspond to the j^{th} restaurant-specific set of table counts for each considered dish and served dish, and override variables of the CRF with loyal customers, respectively. The Gibbs sampler of Algorithm 9 iterates between sequential sampling of the state z_t for each individual value of t given β and $z_{\backslash t}$, sampling of the auxiliary variables $\bar{m}_{jk}$, m_{jk}, and w_{jt} given $z_{1:T}$ and β, and sampling of β given these auxiliary variables.

■ 3.1.3 Blocked Sampling of State Sequences

The HDP-HMM sequential, direct assignment sampler of Algorithm 9 can exhibit slow mixing rates since global state sequence changes are forced to occur coordinate by coordinate. This phenomenon is explored in [148] for the finite HMM. Although the sticky HDP-HMM reduces the posterior uncertainty caused by fast state-switching explanations of the data, the self-transition bias can cause two continuous and temporally separated sets of observations of a given state to be assigned to two distinct states. See Fig. 3.5 for an example. If this occurs, the high probability of self-transition makes it challenging for the sequential sampler to assign those two examples to a common state.

Alternatively, we propose using a variant of the HMM forward-backward procedure described in Sec. 2.6.1 to harness the Markovian structure and jointly sample the state sequence $z_{1:T}$ given the observations $y_{1:T}$, transition probabilities π_k, and parameters θ_k. To take advantage of this procedure, we now must sample the previously marginalized transition distributions and model parameters. In practice, this requires approximating the countably infinite transition distributions using one of the approaches outlined in Sec. 2.9.2. For this chapter, we choose to use the weak limit approximation because of

```
Given the previous state assignments $z_{1:T}^{(n-1)}$ and global transition distribution $\beta^{(n-1)}$:
  1. Set $z_{1:T} = z_{1:T}^{(n-1)}$ and $\beta = \beta^{(n-1)}$. For each $t \in \{1, \dots, T\}$, sequentially
    (a) Decrement $n_{z_{t-1}z_t}$ and $n_{z_t z_{t+1}}$ and remove $y_t$ from the cached statistics for
        the current assignment $z_t = k$:
          $(\hat{\mu}_k, \hat{\Sigma}_k) \leftarrow (\hat{\mu}_k, \hat{\Sigma}_k) \ominus y_t \qquad \hat{\nu}_k \leftarrow \hat{\nu}_k - 1$
    (b) For each of the $K$ currently instantiated states, determine
          $f_k(y_t) = (\alpha\beta_k + n_{z_{t-1}k}) \left( \frac{\alpha\beta_{z_{t+1}} + n_{kz_{t+1}} + \kappa\delta(k, z_{t+1})}{\alpha + n_{k\cdot} + \kappa} \right) t_{\hat{\nu}_k}(y_t; \hat{\mu}_k, \hat{\Sigma}_k)$
        for $z_{t-1} \neq k$, otherwise see Eq. (A.10). Also determine probability $f_{K+1}(y_t)$
        of a new state $K+1$.
    (c) Sample the new state assignment $z_t$:
          $z_t \sim \sum_{k=1}^{K} f_k(y_t)\delta(z_t, k) + f_{K+1}(y_t)\delta(z_t, K+1)$
        If $z_t = K+1$, increment $K$ and transform $\beta$ as follows. Sample $b \sim \text{Beta}(1, \gamma)$
        and assign $\beta_K \leftarrow b\beta_{\tilde{k}}$ and $\beta_{\tilde{k}} \leftarrow (1-b)\beta_{\tilde{k}}$, where $\beta_{\tilde{k}} = \sum_{k=K+1}^{\infty} \beta_k$.
    (d) Increment $n_{z_{t-1}z_t}$ and $n_{z_t z_{t+1}}$ and add $y_t$ to the cached statistics for the new
        assignment $z_t = k$:
          $(\hat{\mu}_k, \hat{\Sigma}_k) \leftarrow (\hat{\mu}_k, \hat{\Sigma}_k) \oplus y_t \qquad \hat{\nu}_k \leftarrow \hat{\nu}_k + 1$
  2. Fix $z_{1:T}^{(n)} = z_{1:T}$. If there exists a $j$ such that $n_{j\cdot} = 0$ and $n_{\cdot j} = 0$, remove $j$ and
     decrement $K$.
  3. Sample auxiliary variables $\boldsymbol{m}$, $\boldsymbol{w}$, and $\bar{\boldsymbol{m}}$ as follows:
    (a) For each $(j, k) \in \{1, \dots, K\}^2$, set $m_{jk} = 0$ and $n = 0$. For each customer in
        restaurant $j$ eating dish $k$, that is for $n = 1, \dots, n_{jk}$, sample
          $x \sim \text{Ber}\left( \frac{\alpha\beta_k + \kappa\delta(j, k)}{n + \alpha\beta_k + \kappa\delta(j, k)} \right)$
        Increment $n$, and if $x = 1$ increment $m_{jk}$.
    (b) For each $j \in \{1, \dots, K\}$, sample the number of override variables in
        restaurant $j$:
          $w_{j\cdot} \sim \text{Binomial}\left( m_{jj}, \rho(\rho + \beta_j(1-\rho))^{-1} \right)$,
        Set the number of informative tables in restaurant $j$ considering dish $k$ to:
          $\bar{m}_{jk} = \begin{cases} m_{jk}, & j \neq k; \\ m_{jj} - w_{j\cdot}, & j = k. \end{cases}$
  4. Sample the global transition distribution from
          $\beta^{(n)} \sim \text{Dir}(\bar{m}_{\cdot 1}, \dots, \bar{m}_{\cdot K}, \gamma)$
  5. Optionally, resample hyperparameters $\gamma$, $\alpha$, and $\kappa$ as described in Appendix C.
```

Algorithm 9. Direct assignment collapsed Gibbs sampler for the sticky HDP-HMM. The algorithm for the HDP-HMM follows directly by setting $\kappa = 0$. Here, we assume Gaussian observations with a normal-inverse-Wishart prior on the parameters of these distributions (see Appendix A). The $\oplus$ and $\ominus$ operators update cached mean and covariance statistics as assignments are added or removed from a given component.

its simplicity and computational efficiency. That is,

$$\begin{aligned} \beta \mid \gamma &\sim \mathrm{Dir}(\gamma/L, \dots, \gamma/L) \\ \pi_j \mid \alpha, \kappa, \beta &\sim \mathrm{Dir}(\alpha\beta_1, \dots, \alpha\beta_j + \kappa, \dots, \alpha\beta_L), \end{aligned} \tag{3.9}$$

where L is the chosen truncation level.

The Gibbs sampler using blocked resampling of $z_{1:T}$ is outlined in Algorithm 10, with derivations found in Appendix B. A similar sampler has been used for inference in HDP hidden Markov trees [97]. However, this work did not consider the complications introduced by multimodal emissions, which we explore in Sec. 3.3. Recently, a slice sampler, referred to as *beam sampling* [170], has been developed for the HDP-HMM. This sampler harnesses the efficiencies of the forward-backward algorithm without having to fix a truncation level for the HDP. However, as we elaborate upon in Sec. 3.2.1, this sampler can suffer from slower mixing rates than our blocked sampler, which utilizes a fixed-order, weak limit truncation of the HDP-HMM.

■ 3.1.4 Hyperparameters

We treat the hyperparameters in the sticky HDP-HMM as unknown quantities and perform full Bayesian inference over these quantities. This emphasizes the role of the data in determining the number of occupied states and the degree of self-transition bias. Our derivation of sampling updates for the hyperparameters of the sticky HDP-HMM is presented in Appendix C; it roughly follows that of the original HDP-HMM [162]. A key step which simplifies our inference procedure is to note that since we have the deterministic relationships

$$\begin{aligned} \alpha &= (1-\rho)(\alpha+\kappa) \\ \kappa &= \rho(\alpha+\kappa), \end{aligned} \tag{3.10}$$

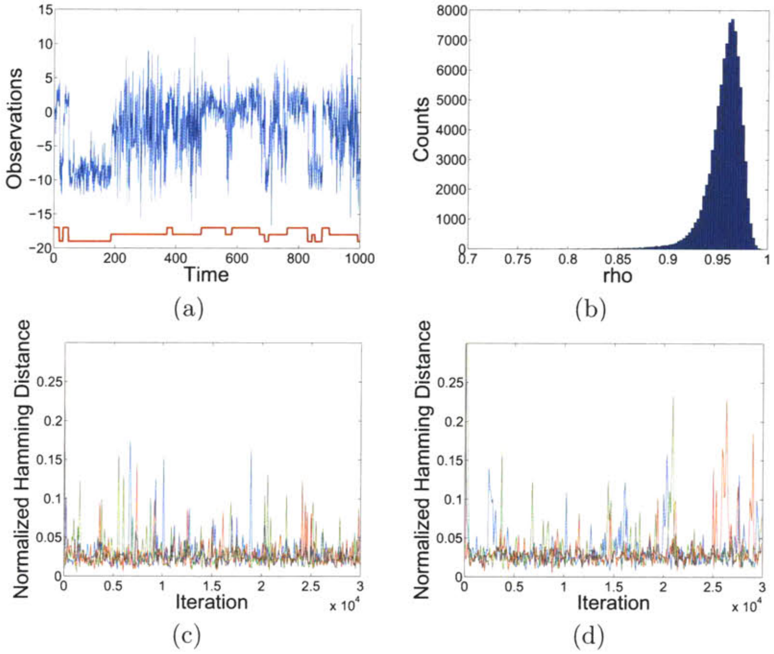
we can treat ρ and $\alpha + \kappa$ as our hyperparameters and sample these values instead of sampling α and κ directly.

■ 3.2 Experiments with Synthetic Data

In this section, we explore the performance of the sticky HDP-HMM relative to the original model (i.e., with the self-transition bias $\kappa = 0$) in a series of experiments with synthetic data. We judge performance according to two metrics: our ability to accurately segment the data according to the underlying state sequence, and the predictive likelihood of held-out data under the inferred model. We additionally empirically assess the improvements in mixing rate achieved by using the blocked sampler of Sec. 3.1.3. The results of Sec. 3.2.1 primarily demonstrate that the sticky HDP-HMM more rapidly finds good segmentations of data with state persistence; in Sec. 3.2.2, the difference in overall modeling power becomes apparent.

```
Given a previous set of state-specific transition probabilities $\boldsymbol{\pi}^{(n-1)}$, the global
transition distribution $\beta^{(n-1)}$, and emission parameters $\boldsymbol{\theta}^{(n-1)}$:
  1. Set $\boldsymbol{\pi} = \boldsymbol{\pi}^{(n-1)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n-1)}$. Working sequentially backwards in time,
     calculate messages $m_{t,t-1}(k)$ :
     (a) For each $k \in \{1, \dots, L\}$, initialize messages to
             $m_{T+1,T}(k) = 1$
     (b) For each $t \in \{T-1, \dots, 1\}$ and for each $k \in \{1, \dots, L\}$, compute
             $m_{t,t-1}(k) = \sum_{j=1}^{L} \pi_k(j) \mathcal{N}(y_t; \mu_j, \Sigma_j) m_{t+1,t}(j)$
  2. Sample state assignments $z_{1:T}$ working sequentially forward in time, starting
     with $n_{jk} = 0$ and $\mathcal{Y}_k = \emptyset$ for each $(j,k) \in \{1, \dots, L\}^2$:
     (a) For each $k \in \{1, \dots, L\}$, compute the probability
             $f_k(y_t) = \pi_{z_{t-1}}(k) \mathcal{N}(y_t; \mu_k, \Sigma_k) m_{t+1,t}(k)$
     (b) Sample a state assignment $z_t$:
             $z_t \sim \sum_{k=1}^{L} f_k(y_t) \delta(z_t, k)$
     (c) Increment $n_{z_{t-1}z_t}$ and add $y_t$ to the cached statistics for the new assignment
         $z_t = k$:
             $\mathcal{Y}_k \leftarrow \mathcal{Y}_k \oplus y_t$
  3. Sample the auxiliary variables $\boldsymbol{m}$, $\boldsymbol{w}$, and $\bar{\boldsymbol{m}}$ as in step 3 of Algorithm 9.
  4. Update the global transition distribution by sampling
             $\beta \sim \mathrm{Dir}(\gamma/L + \bar{m}_{\cdot 1}, \dots, \gamma/L + \bar{m}_{\cdot L})$
  5. For each $k \in \{1, \dots, L\}$, sample a new transition distribution and emission
     parameter based on the sampled state assignments
             $\pi_k \sim \mathrm{Dir}(\alpha\beta_1 + n_{k1}, \dots, \alpha\beta_k + \kappa + n_{kk}, \dots, \alpha\beta_L + n_{kL})$
             $\theta_k \sim p(\theta \mid \lambda, \mathcal{Y}_k)$
     See Appendix B.4.1 for details on resampling $\theta_k$.
  6. Fix $\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}$, $\beta^{(n)} = \beta$, and $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}$.
  7. Optionally, resample hyperparameters $\gamma$, $\alpha$, and $\kappa$ as described in Appendix C.
```

Algorithm 10. Blocked Gibbs sampler for the sticky HDP-HMM. The algorithm for the original HDP-HMM follows directly by setting $\kappa = 0$. Here, we assume Gaussian observations with an independent Gaussian prior on the mean and inverse-Wishart (IW) prior on the covariance (see Appendix B.4.1). The set $\mathcal{Y}_k$ is comprised of the statistics obtained from the observations assigned to state k that are necessary for updating the parameter $\theta_k = \{\mu_k, \Sigma_k\}$. The $\oplus$ operator updates these cached statistics as a new assignment is made.



(a) (b) (c) (d)

Figure 3.6. (a) Observation sequence (blue) and true state sequence (red) for a three-state HMM with state persistence. (b) Histogram of the inferred self-transition proportion parameter, ρ, for the sticky HDP-HMM blocked sampler. (c) Hamming distance over 30,000 Gibbs samples from three chains using the sticky HDP-HMM blocked sampler. (d) Similar plot for the original HDP-HMM.

■ 3.2.1 Gaussian Emissions

We begin our analysis of the sticky HDP-HMM performance by examining a set of simulated data generated from an HMM with Gaussian emissions. The first dataset is generated from an HMM with a high probability of self-transition. The second dataset is from an HMM with a high probability of leaving the current state. In this scenario, our goal is to demonstrate that the sticky HDP-HMM is still able to capture rapid dynamics by inferring a small probability of self-transition.

For all of the experiments with simulated data, we used weakly informative hyperpriors. We placed a Gamma$(1, 0.01)$ prior on the concentration parameters γ and $(\alpha + \kappa)$. The self-transition proportion parameter ρ was given a Beta$(10, 1)$ prior. The parameters of the base measure were set from the data, as will be described for each scenario.

State Persistence

The data for the high persistence case were generated from a three-state HMM with a 0.98 probability of self-transition and equal probability of transitions to the other two states. The observation and true state sequences for the state persistence scenario are

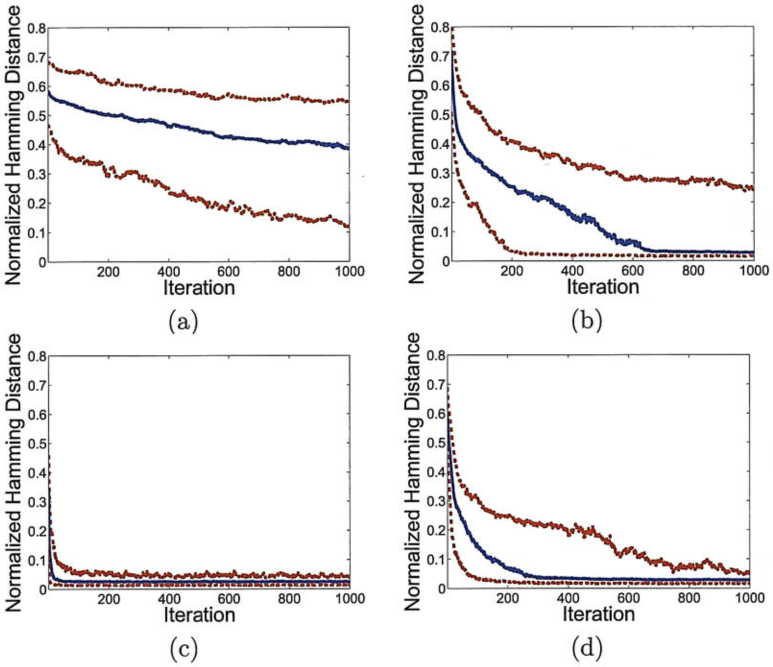
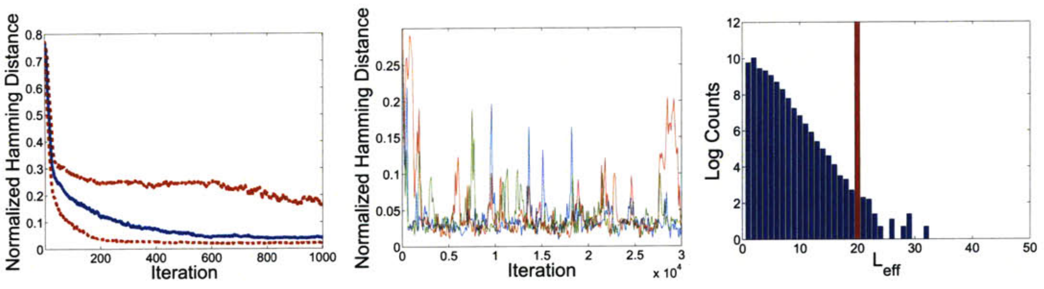


Figure 3.7. For the observation sequence of Fig. 3.6, the median (solid blue) and 10^{th} and 90^{th} quantiles (dashed red) of Hamming distance between the true and estimated state sequences over the first 1,000 Gibbs samples from 200 chains are shown for the (a) sticky HDP-HMM direct assignment sampler, (b) original HDP-HMM direct assignment sampler, (c) sticky HDP-HMM blocked sampler, and (d) original HDP-HMM blocked sampler.

shown in Fig. 3.6(a). We placed a normal inverse-Wishart (NIW) prior (see Sec. 2.4.3) on the space of mean and variance parameters and set the hyperparameters as: 0.01 pseudocounts, mean equal to the empirical mean, three degrees of freedom, and scale matrix equal to 0.75 times the empirical variance. We used this conjugate base measure so that we may directly compare the performance of the blocked and direct assignment samplers of Algorithms 9 and 10. For the blocked sampler, we used a truncation level of $L = 20$.

In Fig. 3.7(a)-(d), we plot the 10^{th}, 50^{th}, and 90^{th} quantiles of the Hamming distance between the true and estimated state sequences over the 1000 Gibbs iterations using the direct assignment and blocked samplers on the sticky and original HDP-HMM models. To calculate the Hamming distance, we used the Munkres algorithm [121] to map the randomly chosen indices of the estimated state sequence to the set of indices that maximize the overlap with the true sequence.

From these plots, we see that the burn-in rate of the blocked sampler using the sticky HDP-HMM is significantly faster than that of any other sampler-model combination. As expected, the sticky HDP-HMM with the sequential, direct assignment sampler (Algorithm 9) gets stuck in state sequence assignments from which it is hard to move



(a) (b) (c)

Figure 3.8. For the beam sampler, we plot: (a) the median (solid blue) and 10^{th} and 90^{th} quantiles (dashed red) of the Hamming distance between the true and estimated state sequences over the first 1,000 Gibbs samples from 200 chains, and (b) the Hamming distance over 30,000 Gibbs samples from three chains. (c) Histogram of the effective beam sampler truncation level, L_{eff}, over the 30,000 Gibbs iterations from the three chains (blue) compared to the fixed truncation level, $L = 20$, used above (red).

away, as conveyed by the flatness of the Hamming error versus iteration number plot in Fig. 3.7(a). For example, the estimated state sequence of Fig. 3.5(b) might have similar parameters associated with states 3, 7, 10 and 11 so that the likelihood is in essence the same as if these states were grouped, but this sequence has a large error in terms of Hamming distance and it would take many iterations to move away from this assignment. Incorporating the blocked sampler with the original HDP-HMM improves the Hamming distance performance relative to the sequential, direct assignment sampler for both the original and sticky HDP-HMM; however, the burn-in rate is still substantially slower than that of the blocked sampler on the sticky model (Algorithm 10).

Recently, a *beam sampling* algorithm [170] has been proposed which adapts slice sampling methods [142] to the HDP-HMM. This approach uses a set of auxiliary slice variables, one for each observation, to effectively truncate the number of state transitions that must be considered at every Gibbs sampling iteration. Dynamic programming methods can then be used to jointly resample state assignments. The beam sampler was inspired by a related approach for DP mixture models [177], which is conceptually similar to retrospective sampling methods [131]. In comparison to our fixed-order, weak limit truncation of the HDP-HMM, the beam sampler provides an asymptotically exact algorithm. However, the beam sampler can be slow to mix relative to our blocked sampler on the fixed, truncated model (see Fig. 3.8 for a comparison on the high persistence dataset examined above.) The issue is that in order to consider a transition which has low prior probability, one needs a correspondingly rare slice variable sample at that time. Thus, even if the likelihood cues are strong, to be able to consider state sequences with several low-prior-probability transitions, one needs to wait for several *rare events* to occur when drawing slice variables. By considering the full, exponentially large set of paths in the truncated state space, we avoid this problem. Of course, the trade-off between the computational cost of the blocked sampler on the fixed, truncated model ($O(TL^2)$) and the slower mixing rate of the beam sampler yields an application-dependent sampler choice.

The Hamming distance plots of Fig. 3.8(a) and (b), when compared to those of Fig. 3.6 and Fig. 3.7, depict the substantially slower mixing rate of the beam sampler than the blocked sampler. However, the theoretical computational benefit of the beam sampler can be seen in Fig. 3.8(c). In this plot, we present a histogram of the effective truncation level, L_{eff}, used over the 30,000 Gibbs iterations on three chains. We computed this effective truncation level by summing over the number of state transitions considered during a full sweep of sampling $z_{1:T}$ and then dividing this number by the length of the dataset, T, and taking the square root. On a more technical note, our fixed, truncated model allows for more vectorization of the code than the beam sampler. Thus, in practice, the difference in computation time between the samplers is significantly less than the $O(L^2/L_{eff}^2)$ factor obtained by counting state transitions.

From this point onwards, we present results only from blocked sampling since we have seen the clear advantages of this method over the sequential, direct assignment sampler.

Fast State-Switching

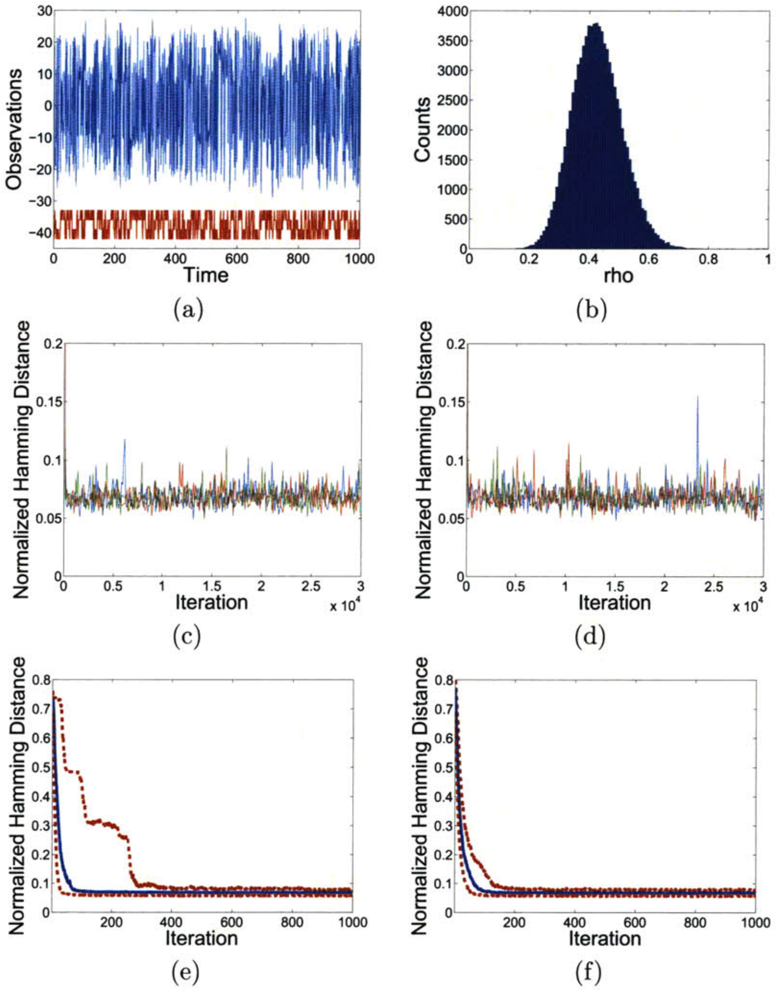
In order to warrant the general use of the sticky model, one would like to know that the incorporated sticky parameter does not preclude learning models with fast dynamics. To this end, we explore the performance of the sticky HDP-HMM on data generated from a model with a high probability of switching between states. Specifically, we generated observations from a four-state HMM with the following transition probability matrix:

$$\begin{bmatrix} 0.4 & 0.4 & 0.1 & 0.1 \\ 0.4 & 0.4 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.4 & 0.4 \\ 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}. \tag{3.11}$$

We once again used a truncation level $L = 20$. Since we are restricting ourselves to the blocked Gibbs sampler, it is no longer necessary to use a conjugate base measure[2]. Instead, we placed an independent Gaussian prior on the mean parameter and an inverse-Wishart prior on the variance parameter. Since we can no longer jointly sample the mean and variance from their posterior, our sampler now relies on iterating between sampling the mean given the variance, and the variance given the mean, each of which yields closed-form posteriors. See Appendix B.4.1 for more details. For the Gaussian prior, we set the mean and variance hyperparameters to be equal to the empirical mean and variance of the entire dataset. The inverse-Wishart hyperparameters were set such that the expected variance is equal to 0.75 times that of the entire dataset, with three degreee of freedom.

The results depicted in Fig. 3.9 confirm that by inferring a small probability of self-transition, the sticky HDP-HMM is indeed able to capture fast HMM dynamics, and just

[2] Conjugate base measures can impose restrictive assumptions, such as the scaling of the variance with the mean in the case of the NIW prior.



(a) (b) (c) (d) (e) (f)

Figure 3.9. (a) Observation sequence (blue) and true state sequence (red) for a four-state HMM with fast state switching. (b) Histogram of the inferred self-transition parameter, ρ, for the sticky HDP-HMM blocked sampler. (c) Hamming distance over 30,000 Gibbs samples from three chains are shown for the sticky HDP-HMM blocked sampler. (d) Similar plot for the original HDP-HMM. The median (solid blue) and 10^{th} and 90^{th} quantiles (dashed red) of Hamming distance between the true and estimated state sequences over the first 1,000 Gibbs samples from 200 chains are shown for the (e) sticky HDP-HMM and (f) original HDP-HMM using the blocked sampler.

as quickly as the original HDP-HMM (although with higher variability.) Specifically, we see that the histogram of the self-transition proportion parameter ρ for this dataset (see Fig. 3.9(b)) is centered around a value close to the true probability of self-transition, which is substantially lower than the mean value of this parameter on the data with high persistence (Fig. 3.6(b).)

3.2.2 Multinomial Emissions

In Fig. 3.6(c)-(d) and Fig. 3.9(c)-(d), we display the Hamming distance associated with three chains over 30,000 Gibbs iterations for both the sticky and original HDP-HMM using blocked sampling on the data of Sec. 3.2.1. From these plots, we do not see the differences between the models that were present at burn-in (see Fig. 3.7). The difference in modeling power, rather than simply burn-in rate, between the sticky and original HDP-HMM is more pronounced when we consider multinomial emissions. This is because the multinomial observations are embedded in a discrete topological space in which there is no concept of similarity between non-identical observation values. In contrast, Gaussian emissions have a continuous range of values in $\mathbb{R}^n$ with a clear notion of *closeness* between observations under the Lebesgue measure, aiding in grouping observations under a single HMM state's Gaussian emission distribution, even in the absence of a self-transition bias.

To demonstrate the increased posterior uncertainty with discrete observations, we generated data from a five-state HMM with multinomial emissions with a 0.98 probability of self-transition and equal probability of transitions to the other four states. The vocabulary, or range of possible observation values, was set to 20. The observation and true state sequences are shown in Fig. 3.10(a). We placed a symmetric Dirichlet prior on the parameters of the multinomial distribution, with the Dirichlet hyperparameters equal to 2 (i.e., $\text{Dir}(2, \dots, 2)$.)

From Fig. 3.10, we see that even after burn-in, many fast-switching state sequences have significant posterior probability under the non-sticky model leading to sweeps through regions of larger Hamming distance error. A qualitative plot of one such inferred sequence after 30,000 Gibbs iterations is shown in Fig. 3.1(c). Such sequences have negligible posterior probability under the sticky HDP-HMM formulation.

In some applications, such as the speaker diarization problem that is explored in Sec. 3.5, one cares about the inferred segmentation of the data into a set of state labels. In this case, the advantage of incorporating the sticky parameter is clear. However, it is often the case that the metric of interest is the predictive power of the learned model, not the accuracy of the inferred state sequence. To study performance under this metric, we simulated 10 test sequences using the same parameters that generated the training sequence. We then computed the likelihood of each of the test sequences under the set of parameters inferred at every 100^{th} Gibbs iteration from iterations 10,000 to 30,000. This likelihood was computed by running the forward-backward algorithm described in Sec. 2.6.1. We plot these results as a histogram in Fig. 3.10(b). From this plot, we see that the fragmentation of data into redundant HMM states can also degrade the predictive performance of the inferred model. Thus, the sticky parameter plays an important role in the Bayesian nonparametric learning of HMMs even in terms of model averaging and predictive power.

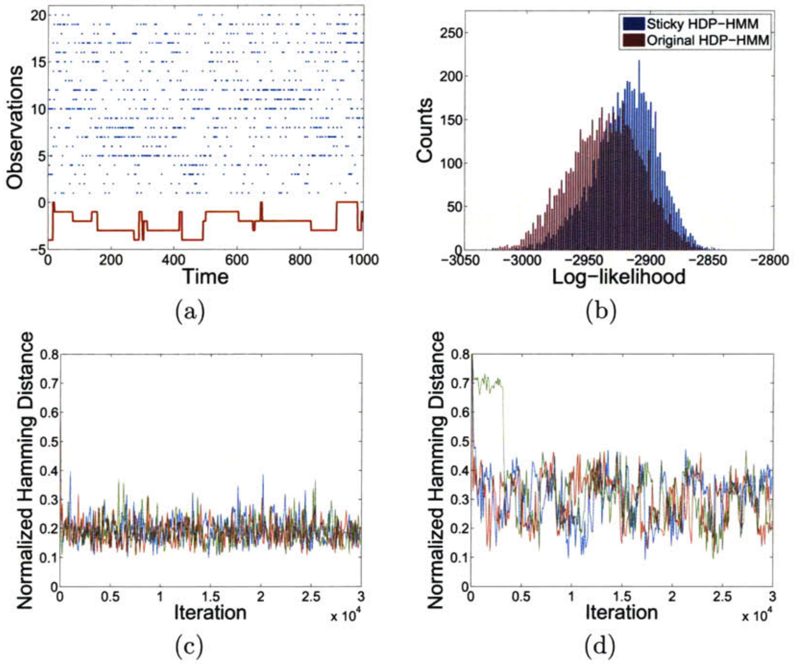


Figure 3.10. (a) Observation sequence (blue) and true state sequence (red) for a five-state HMM with multinomial observations. (b) Histogram of the predictive probability of test sequences using the inferred parameters sampled every 100^{th} iteration from Gibbs iterations 10,000 to 30,000 for the sticky and original HDP-HMM. The Hamming distances over 30,000 Gibbs samples from three chains are shown for the (b) sticky HDP-HMM and (c) original HDP-HMM.

■ 3.2.3 Comparison to Independent Sparse Dirichlet Prior

We have alluded to the fact that the *shared* sparsity of the HDP-HMM induced by β is essential for inferring sparse representations of the data. Although this is clear from the perspective of the prior model, or equivalently the generative process, it is not immediately obvious how much this hierarchical Bayesian constraint helps us in posterior inference. Once we are in the realm of considering a fixed, truncated approximation to the HDP-HMM, one might propose an alternate model in which we simply place a sparse Dirichlet prior, $\text{Dir}(\alpha/L, \dots, \alpha/L)$ with $\alpha/L < 1$, independently on each row of the transition matrix. This is equivalent to setting $\beta = [1/L, \dots, 1/L]$ in the truncated HDP-HMM, which can also be achieved by letting the hyperparameter γ tend to infinity. Indeed, when the data do not exhibit shared sparsity or when the likelihood cues are sufficiently strong, the independent sparse Dirichlet prior model can perform as well as the truncated HDP-HMM. However, in scenarios such as the one depicted in Fig. 3.11, we see substantial differences in performance by considering the HDP-HMM, as well as the inclusion of the sticky parameter. We explore the relative performance of the HDP-HMM and sparse Dirichlet prior model, with and without the sticky parameter, on such a Markov model with multinomial emissions on a vocabulary of size 20.

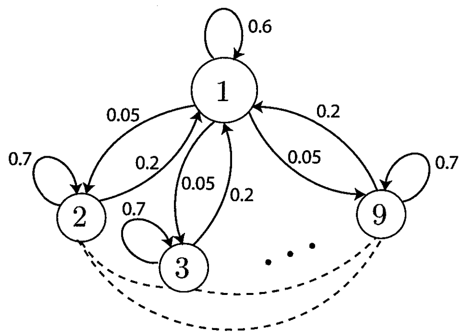


Figure 3.11. State transition diagram for a nine-state HMM with one main state (labeled 1) and eight sub-states (labeled 2 to 9.) All states have a significant probability of self-transition. From the main state, all other states are equally likely. From a sub-state, the most likely non-self-transition is a transition is back to the main state. However, all sub-states have a small probability of transitioning to another sub-state, as indicated by the dashed arcs.

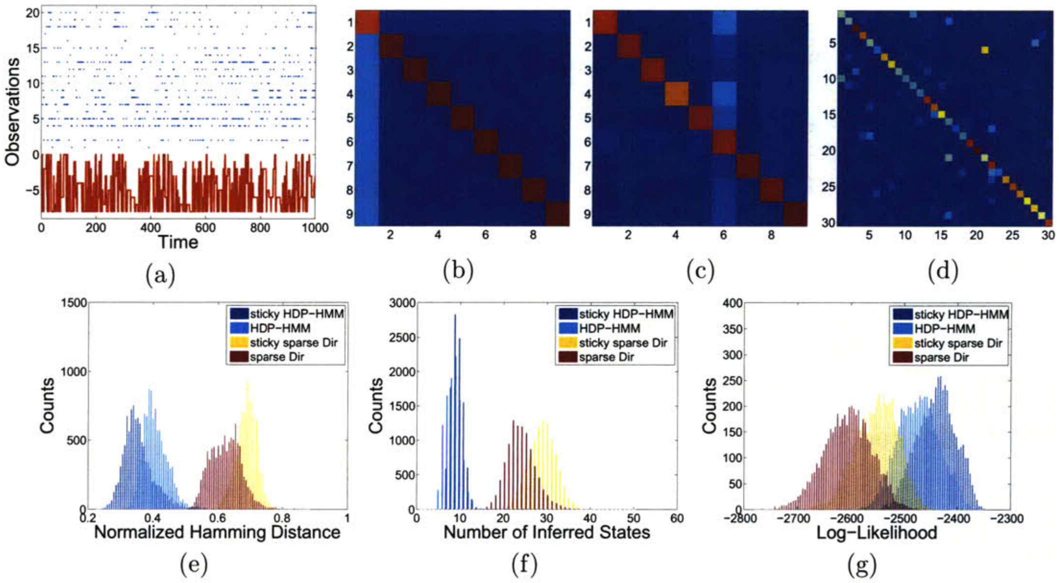
We placed a $\text{Dir}(0.1, \ldots, 0.1)$ prior on the parameters of the multinomial distribution. For the sparse Dirichlet prior model, we assumed a state space of size 50, which is the same as the truncation level we chose for the HDP-HMM (i.e., $L = 50$). The results are presented in Fig. 3.12. From these plots, we see that the hierarchical Bayesian approach of the HDP-HMM does, in fact, improve the learning of a model with shared sparsity. The HDP-HMM consistently infers fewer HMM states and more representative model parameters. As a result, the HDP-HMM has higher predictive likelihood on test data, with an additional benefit gained from using the sticky parameter.

Note that the results of Fig. 3.12(g) also motivate the use of the sticky parameter in the more classical setting of a finite HMM with a standard Dirichlet sparsity prior. A motivating example of the use of sparse Dirichlet priors for finite HMMs is presented in [83].

■ 3.3 Multimodal Emission Densities

In many application domains, the data associated with each hidden state may have a complex, multimodal distribution. We propose to model such emission distributions nonparametrically, using a DP mixture of Gaussians. This formulation is related to the nested DP [143], which uses a Dirichlet process to partition data into groups, and then models each group via a Dirichlet process mixture. The bias towards self-transitions allows us to distinguish between the underlying HDP-HMM states. If the model were free to both rapidly switch between HDP-HMM states and associate multiple Gaussians per state, there would be considerable posterior uncertainty. Thus, as we demonstrate in Sec. 3.4, the sticky parameter is essential in effectively learning such models.

We augment the HDP-HMM state z_t with a term s_t indexing the mixture component of the z_t^{th} emission density. For each HDP-HMM state, there is a unique stick-breaking measure $\psi_k \sim \text{GEM}(\sigma)$ defining the mixture weights of the k^{th} emission density so that



(a) (b) (c) (d)

(e) (f) (g)

Figure 3.12. (a) Observation sequence (blue) and true state sequence (red) for a nine-state HMM with multinomial observations. (b) The true transition probability matrix (TPM). (c)-(d) The inferred TPM at the 30,000th Gibbs iteration for the sticky HDP-HMM and sticky sparse Dirichlet model, respectively, only examining those states with more than 1% of the assignments. For the HDP-HMM and sparse Dirichlet model, with and without the sticky parameter, we plot: (e) the Hamming distance error over 10,000 Gibbs iterations, (f) the inferred number of states with more than 1% of the assignments, and (g) the predictive probability of test sequences using the inferred parameters sampled every 100^{th} iteration from Gibbs iterations 5,000 to 10,000.

$s_t \sim \psi_{z_t}$. Given the augmented state (z_t, s_t), the observation y_t is generated by the Gaussian component with parameter θ_{z_t,s_t}. Note that both the HDP-HMM state index and mixture component index are allowed to take values in a countably infinite set. The generative model is summarized below, and is graphically depicted in Fig. 3.2(b).

$$\begin{array}{rcl}
 & \beta \mid \gamma \sim \mathrm{GEM}(\gamma) & \\
\pi_j \mid \beta, \alpha \sim \mathrm{DP}(\alpha, \beta) & \psi_j \mid \sigma \sim \mathrm{GEM}(\sigma) & j = 1, 2, \ldots \\
 & \theta_{k,j} \mid H, \lambda \sim H(\lambda) & k, j = 1, 2, \ldots \\
z_t | \{\pi_j\}_{j=1}^{\infty}, z_{t-1} \sim \pi_{z_{t-1}} & s_t \mid \{\psi_j\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} & t = 1, \ldots, T \\
 & y_t | \{\theta_{k,j}\}_{k,j=1}^{\infty}, z_t, s_t \sim F(\theta_{z_t,s_t}) & t = 1, \ldots, T.
\end{array} \tag{3.12}$$

■ 3.3.1 Direct Assignment Sampler

Many of the steps of the direct assignment sampler for the sticky HDP-HMM with DP emissions remains the same as for the regular sticky HDP-HMM (Algorithm 9). Specifically, the sampling of the global transition distribution β, the table counts m_{jk} and $\bar{m}_{jk}$, and the override variables w_{jt} are unchanged. The difference arises in how we sample the augmented state (z_t, s_t).

The joint distribution on the augmented state, having marginalized the transition distributions π_k and emission mixture weights ψ_k, is given by

$$p(z_t = k, s_t = j \mid z_{\backslash t}, s_{\backslash t}, y_{1:T}, \beta, \alpha, \sigma, \kappa, \lambda) = p(s_t = j \mid z_t = k, z_{\backslash t}, s_{\backslash t}, y_{1:T}, \sigma, \lambda) \\ p(z_t = k \mid z_{\backslash t}, s_{\backslash t}, y_{1:T}, \beta, \alpha, \kappa, \lambda). \quad (3.13)$$

We then block-sample (z_t, s_t) by first sampling z_t, followed by s_t conditioned on the sampled value of z_t. The term $p(s_t = j \mid z_t = k, z_{\backslash t}, s_{\backslash t}, y_{1:T}, \sigma, \lambda)$ relies on how many observations are currently assigned to the j^{th} mixture component of state k, which we denote by n'_{kj}. These conditional distributions are derived in Appendix A.2 and the resulting Gibbs sampler is outlined in Algorithm 11.

■ 3.3.2 Blocked Sampler

To implement blocked resampling of $(z_{1:T}, s_{1:T})$, we use weak limit approximations to both the HDP-HMM and DP emissions, approximated to levels L and L', respectively. The posterior distributions for β and π_k remain unchanged from the sticky HDP-HMM; that of ψ_k is given by

$$\psi_k \mid z_{1:T}, s_{1:T}, \sigma \sim \text{Dir}(\sigma/L' + n'_{k1}, \dots, \sigma/L' + n'_{kL'}). \quad (3.14)$$

The procedure for sampling the augmented state sequence $(z_{1:T}, s_{1:T})$ is derived in Appendix B.3. The resulting blocked Gibbs sampler for the sticky HDP-HMM with DP emissions is outlined in Algorithm 12.

■ 3.4 Assessing the Multimodal Emissions Model

In this section, we evaluate the ability of the sticky HDP-HMM to infer multimodal emission distributions relative to the model without the sticky parameter. We then apply this extended sticky HDP-HMM to the speaker diarization task.

■ 3.4.1 Mixture of Gaussian Emissions

To test the model of Sec. 3.3, we generated data from a five-state HMM, where the number of Gaussian mixture components for each emission distribution was chosen randomly from a uniform distribution on $\{1, 2, \dots, 10\}$. Each component of the mixture was equally weighted and the probability of self-transition was set to 0.98, with equal probabilities of transitions to the other states. The large probability of self-transition is what disambiguates this process from one with many more HMM states, each with a single Gaussian emission distribution. The resulting observation and true state sequences are shown in Fig. 3.13(a) and (b).

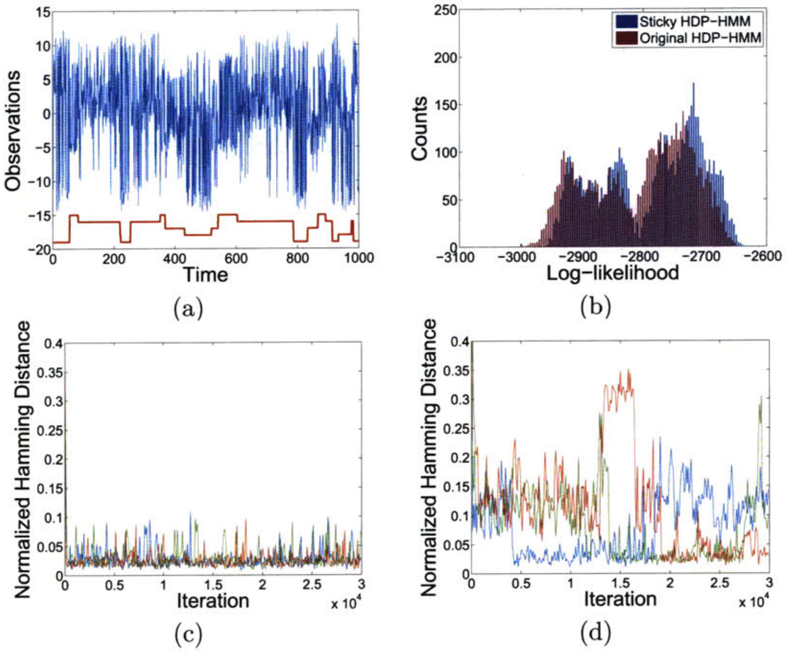
We once again used a non-conjugate base measure and placed a Gaussian prior on the mean parameter and an independent inverse-Wishart prior on the variance parameter of each Gaussian mixture component. The hyperparameters for these distributions

```
Given a previous set of augmented state assignments $(z_{1:T}^{(n-1)}, s_{1:T}^{(n-1)})$ and the global transition distribution $\beta^{(n-1)}$:
1. Set $(z_{1:T}, s_{1:T}) = (z_{1:T}^{(n-1)}, s_{1:T}^{(n-1)})$ and $\beta = \beta^{(n-1)}$. For each $t \in \{1, \dots, T\}$,
   (a) Decrement $n_{z_{t-1}z_t}$, $n_{z_t z_{t+1}}$, and $n'_{z_t s_t}$ and remove $y_t$ from the cached statistics
       for the current assignment $(z_t, s_t) = (k, j)$:
           $(\hat{\mu}_{k,j}, \hat{\Sigma}_{k,j}) \leftarrow (\hat{\mu}_{k,j}, \hat{\Sigma}_{k,j}) \ominus y_t \qquad \hat{\nu}_{k,j} \leftarrow \hat{\nu}_{k,j} - 1$
   (b) For each of the $K$ currently instantiated HDP-HMM states, compute
       i. The predictive conditional distribution for each of the $K'_k$ currently
          instantiated mixture components associated with this HDP-HMM state
              $f'_{k,j}(y_t) = \left(\frac{n'_{kj}}{\sigma + n'_{k\cdot}}\right) t_{\hat{\nu}_{k,j}}(y_t; \hat{\mu}_{k,j}, \hat{\Sigma}_{k,j})$
          and for a new mixture component $K'_k + 1$
              $f'_{k,K'_k+1}(y_t) = \left(\frac{\sigma}{\sigma + n'_{k\cdot}}\right) t_{\hat{\nu}_0}(y_t; \hat{\mu}_0, \hat{\Sigma}_0).$
       ii. The predictive conditional distribution of the HDP-HMM state without
           knowledge of the current mixture component
   $f_k(y_t) = (\alpha\beta_k + n_{z_{t-1}k}) \left(\frac{\alpha\beta_{z_{t+1}} + n_{kz_{t+1}} + \kappa\delta(k, z_{t+1})}{\alpha + n_{k\cdot} + \kappa}\right) \left(\sum_{j=1}^{K'_k} f'_{k,j}(y_t) + f'_{k,K'_k+1}(y_t)\right)$
           for $z_{t-1} \neq k$, otherwise see Appendix A.2. Repeat this procedure for a
           new HDP-HMM state $K+1$ with $K'_{K+1}$ initialized to 0.
   (c) Sample the new augmented state assignment $(z_t, s_t)$ by first sampling $z_t$:
           $z_t \sim \sum_{k=1}^{K} f_k(y_t)\delta(z_t, k) + f_{K+1}(y_t)\delta(z_t, K+1).$
       Then, conditioned on a new assignment $z_t = k$, sample $s_t$:
           $s_t \sim \sum_{j=1}^{K'_k} f'_{k,j}(y_t)\delta(s_t, j) + f'_{k,K'_k+1}(y_t)\delta(s_t, K'_k + 1).$
       If $k = K+1$, increment $K$ and transform $\beta$ as follows. Sample $b \sim \text{Beta}(1, \gamma)$
       and assign $\beta_K \leftarrow b\beta_{\tilde{k}}$ and $\beta_{\tilde{k}} \leftarrow (1-b)\beta_{\tilde{k}}$. If $s_t = K'_k + 1$, then increment $K'_k$.
   (d) Increment $n_{z_{t-1}z_t}$, $n_{z_t z_{t+1}}$, and $n'_{z_t s_t}$ and add $y_t$ to the cached statistics for
       the new assignment $(z_t, s_t) = (k, j)$:
           $(\hat{\mu}_{k,j}, \hat{\Sigma}_{k,j}) \leftarrow (\hat{\mu}_{k,j}, \hat{\Sigma}_{k,j}) \oplus y_t \qquad \hat{\nu}_{k,j} \leftarrow \hat{\nu}_{k,j} + 1$
2. Fix $(z_{1:T}^{(n)}, s_{1:T}^{(n)}) = (z_{1:T}, s_{1:T})$. If there exists a $k$ such that $n_{k\cdot} = 0$ and $n_{\cdot k} = 0$,
   remove $k$ and decrement $K$. Similarly, if there is a $(k, j)$ such that $n'_{kj} = 0$ then
   remove $j$ and decrement $K'_k$.
3. Sample auxiliary variables $\boldsymbol{m}$, $\boldsymbol{w}$, and $\bar{\boldsymbol{m}}$ as in step 3 of Algorithm 9.
4. Sample the global transition distribution $\beta^{(n)}$ as in step 4 of Algorithm 9.
5. Optionally, resample hyperparameters $\sigma$, $\gamma$, $\alpha$, and $\kappa$ as described in App. C.
```

Algorithm 11. Direct assignment collapsed Gibbs sampler for the sticky HDP-HMM with DP emissions.

```
Given a previous set of state-specific transition probabilities $\boldsymbol{\pi}^{(n-1)}$, emission mixture weights $\boldsymbol{\psi}^{(n-1)}$, global transition distribution $\beta^{(n-1)}$, and emission parameters $\boldsymbol{\theta}^{(n-1)}$:
 1. Set $\boldsymbol{\pi} = \boldsymbol{\pi}^{(n-1)}$, $\boldsymbol{\psi} = \boldsymbol{\psi}^{(n-1)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n-1)}$. Working sequentially backwards in time, calculate messages $m_{t,t-1}(k)$ :
    (a) For each $k \in \{1, \dots, L\}$, initialize messages to
            $m_{T+1,T}(k) = 1$
    (b) For each $t \in \{T-1, \dots, 1\}$ and for each $k \in \{1, \dots, L\}$, compute
            $m_{t,t-1}(k) = \sum_{i=1}^{L} \sum_{\ell=1}^{L'} \pi_k(i)\psi_i(\ell)\mathcal{N}(y_{t+1}; \mu_{i,\ell}, \Sigma_{i,\ell})m_{t+1,t}(i)$
 2. Sample augmented state assignments $(z_{1:T}, s_{1:T})$ working sequentially forward in time. Start with $n_{ik} = 0$, $n'_{kj} = 0$, and $\mathcal{Y}_{k,j} = \emptyset$ for $(i,k) \in \{1, \dots, L\}^2$ and $(k,j) \in \{1, \dots, L\} \times \{1, \dots, L'\}$.
    (a) For each $(k,j) \in \{1, \dots, L\} \times \{1, \dots, L'\}$, compute the probability
            $f_{k,j}(y_t) = \pi_{z_{t-1}}(k)\psi_k(j)\mathcal{N}(y_t; \mu_{k,j}, \Sigma_{k,j})m_{t+1,t}(k)$
    (b) Sample an augmented state assignment $(z_t, s_t)$:
            $(z_t, s_t) \sim \sum_{k=1}^{L} \sum_{j=1}^{L'} f_{k,j}(y_t)\delta(z_t, k)\delta(s_t, j)$
    (c) Increment $n_{z_{t-1}z_t}$ and $n'_{z_t s_t}$ and add $y_t$ to the cached statistics for the new assignment $(z_t, s_t) = (k, j)$:
            $\mathcal{Y}_{k,j} \leftarrow \mathcal{Y}_{k,j} \oplus y_t$
 3. Sample the auxiliary variables $\boldsymbol{m}$, $\boldsymbol{w}$, and $\bar{\boldsymbol{m}}$ as in step 3 of Algorithm 9.
 4. Update the global transition distribution $\beta$ as in step 4 of Algorithm 10.
 5. For each $k \in \{1, \dots, L\}$,
    (a) Sample a new transition distribution $\pi_k$ and emission mixture weights $\psi_k$:
            $\pi_k \sim \mathrm{Dir}(\alpha\beta_1 + n_{k1}, \dots, \alpha\beta_k + \kappa + n_{kk}, \dots, \alpha\beta_L + n_{kL})$
            $\psi_k \sim \mathrm{Dir}(\sigma/L' + n'_{k1}, \dots, \sigma/L' + n'_{kL'})$
    (b) For each $j \in \{1, \dots, L'\}$, sample the parameters associated with the $j^{th}$ mixture component of the $k^{th}$ emission distribution:
            $\theta_{k,j} \sim p(\theta \mid \lambda, \mathcal{Y}_{k,j})$
        See Appendix B.4.1 for details on resampling $\theta_{k,j}$.
 6. Fix $\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}$, $\boldsymbol{\psi}^{(n)} = \boldsymbol{\psi}$, $\beta^{(n)} = \beta$, and $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}$.
 7. Optionally, resample hyperparameters $\sigma$, $\gamma$, $\alpha$, and $\kappa$ as described in App. C.
```

Algorithm 12. Blocked Gibbs sampler for the sticky HDP-HMM with DP emissions. Here, we use an independent Gaussian prior on the mean and inverse-Wishart (IW) prior on the covariance (see Appendix B.4.1). The set $\mathcal{Y}_{k,j}$ is comprised of the statistics obtained from the observations assigned to augmented state (k,j) that are necessary for updating the parameter $\theta_{k,j} = \{\mu_{k,j}, \Sigma_{k,j}\}$. The $\oplus$ operator updates these cached statistics as a new assignment is made.



(a) (b) (c) (d)

Figure 3.13. (a) Observation sequence (blue) and true state sequence (red) for a five-state HMM with mixture of Gaussian observations. The Hamming distance over 30,000 Gibbs samples from three chains are shown for the (b) sticky HDP-HMM and (c) original HDP-HMM, both with DP emissions. (d) Histogram of the predictive probability of test sequences using the inferred parameters sampled every 100^{th} iteration from Gibbs iterations 10,000 to 30,000 for the sticky and original HDP-HMM.

were set from the data in the same manner as in the fast-switching scenario. Consistent with the sticky HDP-HMM concentration parameters γ and $(\alpha + \kappa)$, we placed a weakly informative Gamma$(1, 0.01)$ prior on the concentration parameter σ of the DP emissions. All results are for the blocked sampler with truncation levels $L = L' = 20$.

In Fig. 3.13, we compare the performance of the sticky HDP-HMM with DP emissions to that of the original HDP-HMM with DP emissions (i.e., DP emissions, but no bias towards self-transitions.) As with the multinomial observations, when the distance between observations does not directly factor into the grouping of observations into HMM states, there is a considerable amount of posterior uncertainty in the underlying HMM state. Even after 30,000 Gibbs samples, there are still state sequence sample paths with very rapid dynamics. The result of this fragmentation into redundant states is a slight reduction in predictive performance on test sequences, as in the multinomial emission case. See Fig. 3.13(b).

3.5 Speaker Diarization

The *speaker diarization* task involves segmenting an audio recording into speaker-homogeneous regions, while simultaneously identifying the number of speakers. We tested the performance of the sticky HDP-HMM with DP emissions[3] for this task on the data distributed by NIST as part of the Rich Transcription 2004-2007 meeting recognition evaluations [126]. We used the first 19 Mel Frequency Cepstral Coefficients (MFCCs), computed over a 30ms window every 10ms, as our feature vector. When working with this dataset, we discovered that: (1) the high frequency content of these features contained little discriminative information, and (2) without a minimum speaker duration, the sticky HDP-HMM inferred within speaker dynamics in addition to global speaker changes. To jointly address these issues, we defined the observations as averages over 250ms, non–overlapping blocks. A minimum speaker duration of 500ms was set by associating two of these observations with each hidden state. To regularize the learning in this high-dimensional space, we also tied the covariances of within–state mixture components (i.e., each speaker–specific mixture component was forced to have identical covariance structure.) See Fig. 3.14 for a block diagram of the processing of audio into diarizations.

As in Sec. 3.4.1, we used a non-conjugate prior on the mean and covariance parameters of the Gaussian components of the mixture emissions. Specifically, we placed a normal prior on the mean parameter with mean equal to the empirical mean and covariance equal to 0.75 times the empirical covariance, and an inverse-Wishart prior on the covariance parameter with 1000 degrees of freedom and expected covariance equal to the empirical covariance. For the concentration parameters, we placed a Gamma$(12, 2)$ prior on γ, a Gamma$(6, 1)$ prior on $\alpha + \kappa$, and a Gamma$(1, 0.5)$ prior on σ. The self-transition proportion parameter ρ was given a Beta$(500, 5)$ prior. For each of the 21 meetings, we ran 10 chains of the blocked Gibbs sampler of Algorithm 12 for 10,000 iterations for both the original and sticky HDP-HMM with DP emissions.

For the NIST speaker diarization evaluations, the goal is to produce a single segmentation for each meeting. Due to the label-switching issue (i.e., under our exchangeable prior, labels are arbitrary entities that do not necessarily remain consistent over Gibbs iterations), we cannot simply integrate over multiple Gibbs sampled state sequences. We propose two solutions to this problem. The first is to simply choose from a fixed set of Gibbs samples the one that produces the largest likelihood given the estimated parameters (marginalizing over state sequences), and then produce the corresponding Viterbi state sequence. This heuristic, however, is sensitive to over-fitting and will, in general, be biased towards solutions with more states.

An alternative, and more robust metric is what we refer to as the *minimum expected Hamming distance.* We first choose a large reference set $\mathcal{R}$ of state sequences produced by the Gibbs sampler and a possibly smaller set of test sequences $\mathcal{T}$. Then,

[3] Although not presented here, the sticky HDP-HMM with single Gaussian emissions performed significantly worse. It is well-established within the speech community that speaker-specific emissions are multimodal, and thus models typically employ mixture of Gaussian emissions [185].

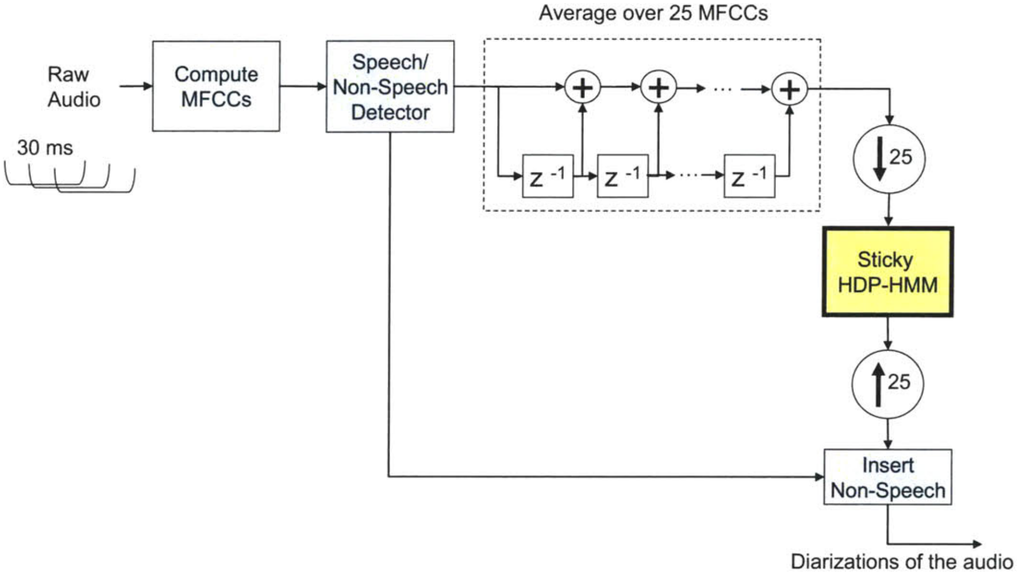


Figure 3.14. Speaker diaraization block diagram. Raw audio is processed into Mel Frequency Cepstral Coefficients (MFCCs) over 30 ms windows every 10 ms. The MFCCs are then separated into speech and non-speech feature sequences. The results of these two steps are provided by ICSI using the algorithms in [185]. We then average non-overlapping blocks of 25 speech features which are passed to the sticky HDP-HMM with DP emissions model. The state sequences produced by the Gibbs sampler of Algorithm 12 are upsampled and the non-speech sequence inserted to produce the final diarizations.

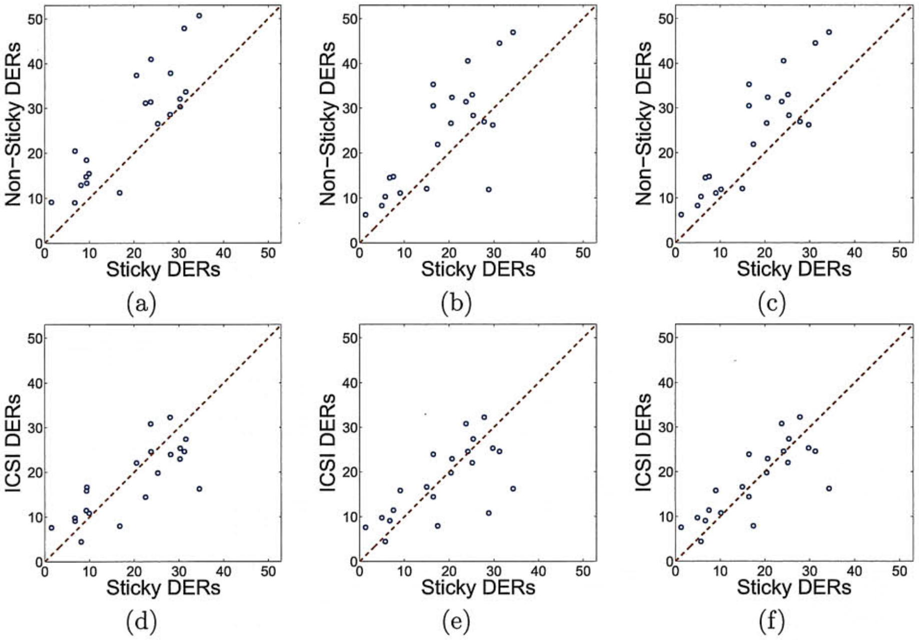
for each sequence i in the test set $\mathcal{T}$, we compute the empirical mean Hamming distance between the test sequence and the sequences in the reference set $\mathcal{R}$; we denote this empirical mean by $\hat{H}_i$. We then choose the test sequence j^* that minimizes this expected Hamming distance. That is,

$$j^* = \arg\min_{i \in \mathcal{T}} \hat{H}_i.$$

The empirical mean Hamming distance $\hat{H}_i$ is a *label-invariant loss function* since it does not rely on labels remaining consistent across samples—we simply compute

$$\hat{H}_i = \frac{1}{|\mathcal{R}|} \sum_{j \in \mathcal{R}} \text{Hamm}(i, j),$$

where $\text{Hamm}(i, j)$ is the Hamming distance between sequences i and j after finding the optimal permutation of the labels in test sequence i to those in reference sequence j. At a high level, this method for choosing state sequence samples aims to produce segmentations of the data that are *typical* samples from the posterior. Asymptotically, this approach will minimize a posterior expected risk. Jasra et al. [78] provides an overview of some related techniques to address the label-switching issue. Although we



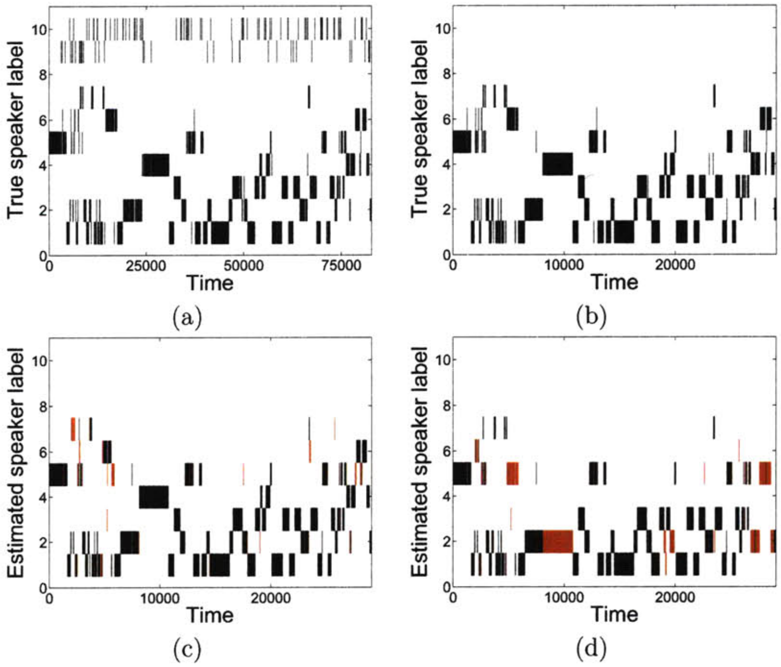
(a) (b) (c)

(d) (e) (f)

Figure 3.15. (a)-(c) For each of the 21 meetings, comparison of diarizations using sticky vs. original HDP-HMM with DP emissions. In (a) we plot the DERs corresponding to the Viterbi state sequence using the parameters inferred at Gibbs iteration 10,000 that maximize the likelihood, and in (b) the DERs using the state sequences that minimize the expected Hamming distance. Plot (c) is the same as (b), except for running the 10 chains for meeting 16 out to 50,000 iterations. (d)-(f) Comparison of the sticky HDP-HMM with DP emissions to the ICSI errors under the same conditions.

could have chosen any label-invariant loss function to minimize, we chose the Hamming distance metric because it is closely related to the official NIST *diarization error rate* (DER) that is calculated during the evaluations. The final metric by which the speaker diarization algorithms are judged is the *overall* DER, a weighted average over the set of meetings based on the length of each meeting.

In Fig. 3.15(a), we report the DER of the chain with the largest likelihood given the parameters estimated at the $10,000^{th}$ Gibbs iteration for each of the 21 meetings, comparing the sticky and original HDP-HMM with DP emissions. The sticky model's temporal smoothing provides substantial performance gains. Although not depicted here, the likelihoods given the parameter estimates under the original HDP-HMM are almost always higher than those under the sticky model. This phenomenon is due to the fact that without the sticky parameter, the HDP-HMM over-segments the data and thus produces parameter estimates more finely tuned to the data resulting in higher likelihoods. Since the original HDP-HMM is contained within the class of sticky models (i.e., when $\kappa = 0$), there is some probability that state sequences similar to those under the original model will eventually arise using the sticky model. Thus, the likelihood metric is not very robust as one would expect the performance under the sticky model



(a)

(b)

(c)

(d)

Figure 3.16. (a) True state sequence for the NIST_20051102-1323 meeting (meeting 16), with labels 9 and 10 indicating times of overlapping- and non- speech, respectively, missed by the speech/non-speech preprocessor. (b) True state sequence with the overlapping- and non- speech time steps removed. (c)-(d) Plotted only over the time-steps as in (b), the state sequences inferred by the sticky HDP-HMM with DP emissions at Gibbs iteration 10,000 chosen using the most likely and minimum expected Hamming distance metrics, respectively. Incorrect labels are shown in red.

to degrade given enough Gibbs chains and/or iterations. In Fig. 3.15(b), we instead report the DER of the chain whose state sequence estimate at Gibbs iteration 10,000 minimizes the expected Hamming distance to the sequences estimated every 100 Gibbs iteration, discarding the first 5,000 iterations. Due to the slow mixing rate of the chains in this application, we additionally discard samples whose normalized log-likelihood is below 0.1 units of the maximum at Gibbs iteration 10,000. For the sticky HDP-HMM, this results in using an average of 270 samples per meeting and, for the original HDP-HMM, 464 samples. The disparity in these numbers arises from the fact that, as further discussed in Sec. 3.6, the sticky model has more chains that have not mixed by 5,000 iterations. From this figure, we see that the sticky model still significantly outperforms the original HDP-HMM, implying that most state sequences produced by the original model are worse, not just the one corresponding to the most-likely sample. One noticeable exception to this trend is the NIST_20051102-1323 meeting (meeting 16). For the sticky model, the state sequence using the maximum likelihood metric had very low DER (see Fig. 3.16(c)); however, there were many chains that merged speakers and produced segmentations similar to the one in Fig. 3.16(d), resulting in such a

Overall DERs (%)	Min Hamming	Max Likelihood	2-Best	5-Best
Sticky HDP-HMM	19.01 (17.84)	19.37	16.97	14.61
Non-Sticky HDP-HMM	23.91	25.91	23.67	21.06

Table 3.1. Overall DERs for the sticky and original HDP-HMM with DP emissions using the minimum expected Hamming distance and maximum likelihood metrics for choosing state sequences at Gibbs iteration 10,000. For the maximum likelihood criterion, we show the best overall DER if we consider the top two or top five most-likely candidates. The number in the parentheses is the performance when running meeting 16 for 50,000 Gibbs iterations. The overall ICSI DER is 18.37%.

sequence minimizing the expected Hamming distance. See Sec. 3.6 for a discussion on the issue of merged speakers. Running meeting 16 for 50,000 Gibbs iterations improved the performance, as depicted by the revised results in Fig. 3.15(c). We summarize our overall performance in Table 3.1, and note that (when using the 50,000 Gibbs iterations for meeting 16) we obtain an overall DER of 17.84% using the sticky HDP-HMM versus the 23.91% of the original HDP-HMM model.

As a further comparison, the ICSI team's algorithm [185], by far the best performer at the 2007 competition, has an overall DER of 18.37%. The ICSI team's algorithm uses agglomerative clustering, and requires significant tuning of parameters on representative training data. In contrast, our hyperparameters are automatically set meeting-by-meeting, as outlined at the beginning of this section. An additional benefit of the sticky HDP-HMM over the ICSI approach is the fact that there is inherent posterior uncertainty in this task, and by taking a Bayesian approach we are able to provide several interpretations. When considering the best per-meeting DER for the five most likely samples, our overall DER drops to 14.61% (see Table 3.1). Meeting VT_20050304-1300 meeting (meeting 18) is an example for which providing multiple solutions is helpful. For this meeting, multiple chains provided segmentations similar to the one depicted in Fig. 3.17(b) with speaker 2 split into two inferred speakers. Examining the observations from this meeting, speaker 2 looks noticeably different over time. However, another chain produced a segmentation which correctly grouped all of the times corresponding to this speaker (see Fig. 3.17(a)). Although not useful in the NIST evaluations, providing multiple segmentations could prove of importance in practice.

To ensure a fair comparison on this dataset, we use the same speech/non-speech pre-processing as ICSI, so that the differences in our performance are due to changes in the identified speakers. Non-speech refers to time when nobody is speaking. The pre-processing step of removing non-speech observations is important in ensuring that the learned acoustic models are not corrupted by non-speech information. As depicted in Fig. 3.18, both our performance and that of ICSI depend significantly on the quality of this pre-processing step. In Fig. 3.18(a), we compare the meeting-by-meeting DERs of the sticky HDP-HMM, original HDP-HMM, and ICSI algorithm, and in Fig. 3.18(b) we plot the fraction of post-processed data that still contains overlapping- and non-

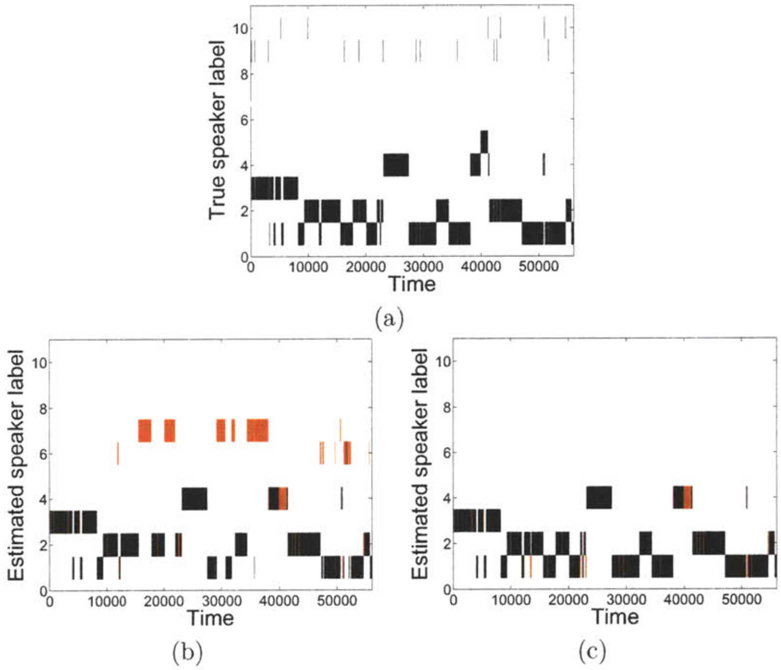


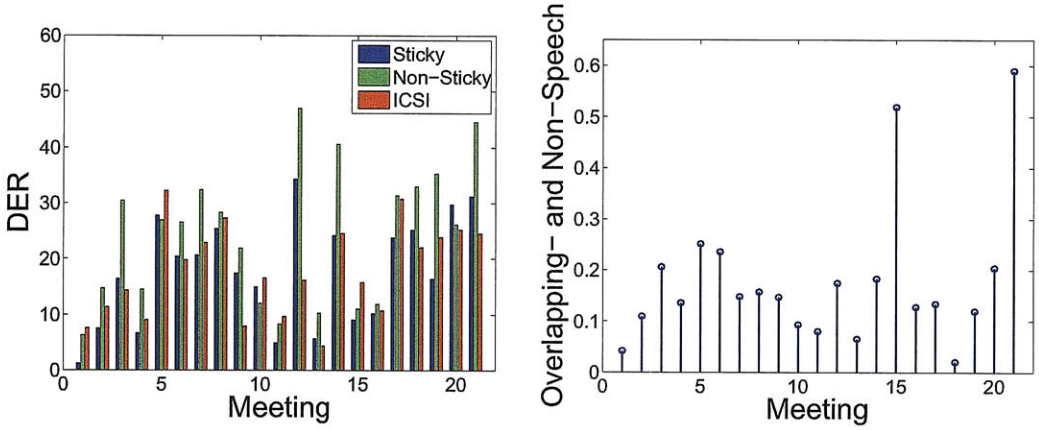
Figure 3.17. (a) True state sequence for the VT_20050304-1300 meeting (meeting 18), with labels 9 and 10 indicating times of overlapping- and non- speech, respectively, missed by the speech/non-speech preprocessor. (b) State sequence inferred by the sticky HDP-HMM with DP emissions at Gibbs iteration 10,000 chosen using the most likely metric. The corresponding DER is 20.48%. (c) State sequence at Gibbs iteration 10,000 for the chain ranked fifth according to the likelihood metric. The corresponding DER is 4.81%. Compare to ICSI's DER of 22.00% for this meeting. .

speech[4]. It is clear from Fig. 3.18(a) that the sticky HDP-HMM with DP emissions provides performance comparable to that of the ICSI algorithm while the original HDP-HMM with DP emissions performs significantly worse. Overall, the results presented in this section demonstrate that the sticky HDP-HMM with DP emissions provides an elegant and empirically effective speaker diarization method.

■ 3.6 Discussion and Future Work

We have examined some of the limitations of the HDP-HMM presented in [162], and demonstrated the considerable benefits of a sticky HDP-HMM in which a separate parameter captures state persistence. By developing a hierarchical Bayesian approach in which a prior is placed on this state persistence parameter, this sticky HDP-HMM provides a general solution that remains capable of capturing fast dynamics. One of the main contributions of the work presented in this chapter is fully integrating this

[4] Not shown in this plot is the amount of actual speech removed by the speech/non-speech preprocessor.



(a)

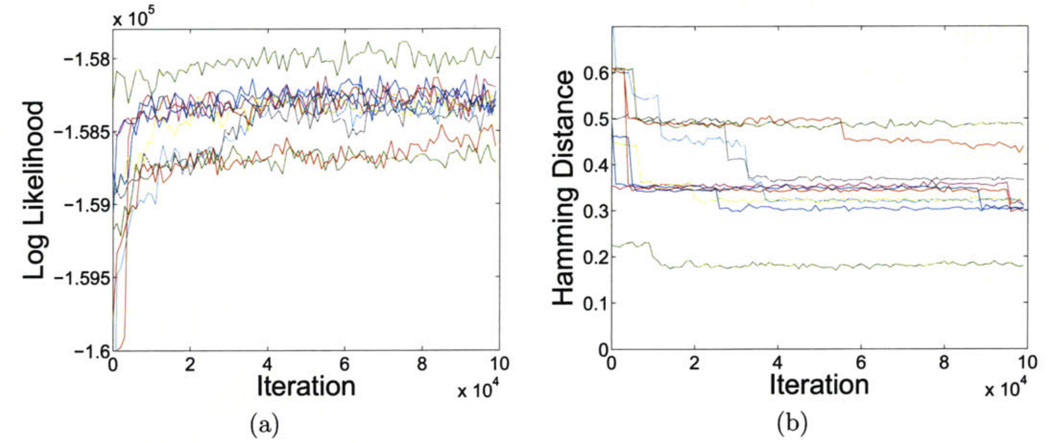
(b)

Figure 3.18. (a) Chart comparing the DERs of the sticky and original HDP-HMM with DP emissions to those of ICSI for each of the 21 meetings. Here, we chose the state sequence at the $10,000^{th}$ Gibbs iteration that minimizes the expected Hamming distance. For meeting 16 using the sticky HDP-HMM with DP emissions, we chose between state sequences at Gibbs iteration 50,000. (b) Plot of the fraction of overlapping- or non- speech in the post-processed data for each of the 21 meetings.

prior within the Bayesian nonparametric framework. For concise presentation, these derivations were extracted to the appendices.

We have also shown that this sticky HDP-HMM allows a fully Bayesian nonparametric treatment of multimodal emissions, disambiguated by its bias towards self-transitions. Accommodating multimodal emissions is a necessary step in describing the data found in many real-world applications, such as the speaker diarization task of Sec. 3.5. Without providing any application-specific information, the sticky HDP-HMM with DP emissions yields a speaker diarization algorithm strongly competitive with the current state of the art. This application, along with extensive testing on synthetic data, clearly demonstrate the practical importance of our extensions.

Finally, we presented efficient sampling techniques with mixing rates that improve on the state of the art by harnessing the Markovian structure of the HDP-HMM. Specifically, we proposed employing a truncated approximation to the HDP and block-sampling the state sequence using a variant of the forward-backward algorithm. Although the blocked samplers of Algorithms 10 and 12 yield substantially improved mixing rates over the sequential, direct assignment samplers of Algorithms 9 and 11, there are still some pitfalls to these sampling methods. One issue is that for each new considered state, the parameter sampled from the prior distribution must better explain the data than the parameters associated with other states that have already been informed by the data. In high-dimensional applications, and in cases where state-specific emission distributions are not clearly distinguishable, this method for adding new states poses a significant challenge. The data in the speaker diarization task is



(a) (b)

Figure 3.19. Trace plots of (a) log-likelihood and (b) Hamming distance error for 10 chains over 100,000 Gibbs iterations for the NIST_20051102-1323 meeting (meeting 16).

both high-dimensional and often has only marginally distinguishable speakers, leading to extremely slow mixing rates, as indicated by the trace plots in Fig. 3.19 of various indicators such as Hamming distance and log-likelihood for 100,000 Gibbs iterations of meeting 16. Many of our errors in this application can be attributed to merged speakers, as depicted in Fig. 3.16(d). On such large datasets, the computation cost of running hundreds of thousands of Gibbs iterations presents an insurmountable barrier. A direction for future work is to develop split-merge algorithms for the HDP and HDP-HMM similar to those developed in [77] for the DP mixture model.

A limitation of the HMM in general is that the observations are assumed conditionally independent and identically distributed given the state sequence. This assumption is often insufficient in capturing the complex temporal dependencies exhibited in real-world data. In the following chapter, we consider Bayesian nonparametric versions of models better suited to such applications, such as the switching linear dynamical system (SLDS) and switching vector autoregressive (VAR) process. A first attempt at developing such models is presented in [44]. An inspiration for the sticky HDP-HMM actually came from considering the original HDP-HMM as a prior for an SLDS. In such scenarios in which one does not have direct observations of the underlying state sequence, the issues arising from not properly capturing state persistence are exacerbated. The sticky HDP-HMM presented in this chapter provides a more robust building block for developing more complicated Bayesian nonparametric dynamical models.

Chapter 4

Bayesian Nonparametric Learning of SLDS

LINEAR dynamical systems (LDSs) are useful in describing dynamical phenomena as diverse as human motion [133, 140], financial time-series [27, 94, 154], maneuvering targets [43, 145], and the dance of honey bees [129]. However, such phenomena often exhibit structural changes over time, and the LDS models which describe them must also change. For example, a ballistic missile makes an evasive maneuver; a country experiences a recession, a central bank intervention, or some national or global event; a honey bee changes from a *waggle* to a *turn right* dance. Some of these changes will appear frequently, while others are only rarely observed. In addition, there is always the possibility of a new, previously unseen dynamical behavior. These considerations motivate us to develop a Bayesian nonparametric approach for learning *switching* LDS (SLDS) models. We also consider a special case of the SLDS—the switching vector autoregressive (VAR) model—in which direct observations of the underlying dynamical process are assumed available. Although a special case of the general linear systems framework, autoregressive models have simplifying properties that often make them a practical choice in applications.

One can view the SLDS, and the simpler switching VAR process, as an extension of hidden Markov models (HMMs) in which each HMM state, or *mode*, is associated with a linear dynamical process. While the HMM makes a strong Markovian assumption that observations are conditionally independent given the mode, the SLDS and switching VAR processes are able to capture more complex temporal dependencies often present in real data. Most existing methods for learning SLDS and switching VAR processes rely on either fixing the number of HMM modes, such as in [129], or considering a change-point detection formulation where each inferred change is to a new, previously unseen dynamical mode, such as in [188]. In this chapter we show how one can remain agnostic about the number of dynamical modes while still allowing for returns to previously exhibited dynamical behaviors.

As we examined in Chapter 3, the hierarchical Dirichlet process (HDP) can be used as a prior on the parameters of HMMs with unknown mode space cardinality [11, 162]. We proposed a variant on the HDP-HMM—the *sticky HDP-HMM*—that provides improved control over the number of modes inferred; we demonstrate that such control is crucial for the problems we examine in this chapter. Our Bayesian nonparametric approach for learning switching dynamical processes extends the sticky HDP-HMM formulation to learn an unknown number of persistent dynamical modes and thereby capture a wider range of temporal dependencies. We then explore a method for learning which components of the underlying state vector contribute to the dynamics of each mode by employing *automatic relevance determination* (ARD) [9,112,124]. The resulting model allows for learning realizations of SLDS that switch between an unknown number of dynamical modes with possibly varying state dimensions, or switching VAR processes with varying autoregressive orders.

Paoletti et al. [130] provide a survey of recent approaches to identification of switching dynamical models. The typical identification problem for these switching models, in the most general sense, involves learning: (i) the number of dynamical modes, (ii) the model order, and (iii) the associated dynamic parameters. Most approaches assume that the model order is the same for each dynamical mode. For noiseless switching VAR processes, Vidal et al. [174] present an exact algebraic approach. However, the method relies on fixing the maximal mode space cardinality and autoregressive order, which is assumed shared among modes. Additionally, extensions to the noisy case rely on heuristics. Psaradakis and Spagnolo [138] alternatively consider a penalized likelihood approach to identification of stochastic switching VAR processes.

For SLDS, identification is significantly more challenging, and methods typically rely on simplifying assumptions such as deterministic dynamics or knowledge of the mode space. If one knew the mode sequence, then one could partition the data according to the underlying mode sequence and then employ the techniques described in Sec. 2.7.4 for identification of single LDS. However, when addressing the issue of stochastic realization or system identification of SLDS, we assume that the mode sequence is a latent variable of our model. Huang et al. [70] present an approach that embeds the input/output data in a higher-dimensional space and finds the switching times by segmenting the data into distinct subspaces [173]. This algebraic approach assumes deterministic dynamics and claims robustness to moderate amounts of noise. Kotsalis et al. [99] develop a balanced truncation algorithm for SLDS assuming the switches between modes are i.i.d. within a fixed, finite set; the authors also present a method for model-order reduction of HMMs[1]. In [135], a realization theory is presented for what the authors refer to as generalized jump-Markov linear systems (GJMLS) in which the dynamic matrix depends both on the previous mode and current mode. The authors mention that it is unclear whether a similar theory can be developed for the standard SLDS we consider in this chapter. Finally, when the number of dynamical modes is assumed known, Ghahramani and Hinton [52] present a variational approach to segmenting the data into the linear dynamical regimes and learning the associated dynamic parameters[2].

[1] The problem of identification of HMMs is thoroughly analyzed in [4].

[2] The formulation of Ghahramani and Hinton [52] uses a *mixture of experts* SLDS representation in

Many questions of observability and identifiability of SLDS in the absence of noise are addressed in [172]. Specifically, a set of sufficient conditions are provided for the initial continuous state $\boldsymbol{x}_0$ and discrete mode sequence $z_{1:T}$ to be observable given fixed model parameters. The authors argue that if both the dimension d of the continuous state and the number of possible modes K are unconstrained, there is an infinite set of systems that realize the same set of observations $\boldsymbol{y}_{1:T}$ while differing in $\{\boldsymbol{x}_0, z_{1:T}\}$. Even when limiting one of these two degrees of freedom, the problem of realization is ill-posed. In this chapter, we circumvent having to limit both d and K, and having to check the detailed conditions provided in [172], by taking a Bayesian approach. One can interpret the control literature on non-identifiable systems as arising in classical statistics when multiple models have equivalent likelihood[3]. In the Bayesian approach that we adopt, it is the specific prior we place on the parameters that allows us to distinguish between the set of these equivalent models. Our choice of prior penalizes more complicated models, both in terms of the number of modes and the state dimension describing each mode. Thus, instead of placing hard constraints on the model, we simply increase the posterior probability of simpler explanations of the data. As opposed to a penalized likelihood approach using *Akaike's information criterion* (AIC) [3] or the *Bayesian information criterion* (BIC) [147], our approach provides a model complexity penalty in a purely Bayesian manner.

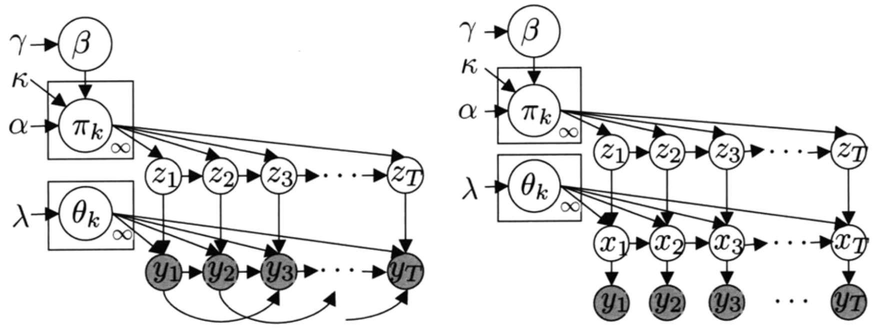
In summary, previous methods for identification of the SLDS we consider in this thesis rely on assuming either: (i) deterministic dynamics, (ii) a fixed number of dynamical modes, or (iii) non-Bayesian penalties on model complexity. The approach we present herein aims to address identification of mode space cardinality and model order of SLDS and switching VAR processes within a Bayesian nonparametric framework. We also demonstrate that allowing for variable order models provides insight into the structure of the underlying phenomenon.

■ 4.1 The HDP-SLDS and HDP-AR-HMM Models

For greater modeling flexibility within the Bayesian framework, we take a nonparametric approach in defining the mode space of our switching dynamical processes. Specifically, we develop extensions of the sticky HDP-HMM of Chapter 3 for both the SLDS and switching VAR models described in Sec. 2.7.3. For the SLDS, we consider conditionally-dependent emissions of which only noisy observations are available (see Fig. 4.1(b).) We refer to this model as the *HDP-SLDS*. The switching VAR(r) process, with r denoting the autoregressive order, can similarly be posed using an HDP-HMM in which the observations are modeled as conditionally VAR(r). This model is referred to as the

which M different continuous-valued state sequences evolve independently with linear dynamics and the Markovian dynamical mode, taking values in $1, \dots, M$, selects which state sequence is observed at a given time.

[3] The definition of identifiability in the control literature differs from that in statistics, though with obvious relations. There is also a concept of Bayesian non-identifiability in which multiple parameters have the same *posterior* probability. See [15, 49].



(a) (b)

Figure 4.1. Grapical models of sticky HDP-HMM prior on switching (a) VAR(2) and (b) SLDS processes with the mode evolving as $z_{t+1}|\{\pi_k\}_{k=1}^{\infty}, z_t \sim \pi_{z_t}$ for $\pi_k|\alpha,\kappa,\beta \sim \text{DP}(\alpha{+}\kappa, (\alpha\beta{+}\kappa\delta_k)/(\alpha{+}\kappa))$. Here, $\beta \mid \gamma \sim \text{GEM}(\gamma)$ and $\theta_k \mid H, \lambda \sim H(\lambda)$. The dynamical processes are as in Eq. (4.1).

HDP-AR-HMM and is depicted in Fig. 4.1(a). The generative processes for these two models are summarized as follows:

	HDP-AR-HMM	HDP-SLDS
Mode dynamics	$z_t \sim \pi_{z_{t-1}}$	$z_t \sim \pi_{z_{t-1}}$
Observation dynamics	$\boldsymbol{y}_t = \sum_{i=1}^{r} A_i^{(z_t)} \boldsymbol{y}_{t-i} + \boldsymbol{e}_t(z_t)$	$\boldsymbol{x}_t = A^{(z_t)} \boldsymbol{x}_{t-1} + \boldsymbol{e}_t(z_t)$
		$\boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{w}_t$

(4.1)

Here, π_j is as defined in the sticky HDP-HMM, namely,

$$\pi_j \mid \alpha, \kappa, \beta \sim \text{DP}\left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}\right). \tag{4.2}$$

The additive noise processes are defined as

$$\boldsymbol{e}_t(k) \sim \mathcal{N}(0, \Sigma^{(k)}) \qquad \boldsymbol{w}_t \sim \mathcal{N}(0, R). \tag{4.3}$$

For the HDP-SLDS, we place a priors on the *dynamic parameters* $\{A^{(k)}, \Sigma^{(k)}\}$ and on measurement noise R and infer their posterior from the data. We do, however, fix the measurement matrix, C, for reasons of identifiability. As given by Eq. (2.155), there exists a set of equivalent systems in terms of the input-output relationship. Let $\tilde{C} \in \mathbb{R}^{d \times n}$, $n \geq d$, be the measurement matrix associated with a dynamical system defined by $\tilde{A}$ and $\tilde{G}$, where we recall from Sec. 2.7.4 that $\tilde{G} = \tilde{A}\tilde{P}_x\tilde{C}^T$ with $\tilde{P}_x$ the steady-state covariance matrix. Assume that $\tilde{C}$ has full row rank. Then, without loss of generality, we may consider $C = [I_d \;\; 0]$ since there exists an invertible transformation T such that the triplet

$$(A = T^{-1}\tilde{A}T, C = \tilde{C}T = [I_d \;\; 0], G = T^{-1}\tilde{G}) \tag{4.4}$$

is in the equivalence class $\mathcal{M}(\tilde{A}, \tilde{C}, \tilde{G})$ as defined in Eq. (2.155). Since the measurement matrix is shared for all modes of the HDP-SLDS, the similarity transformation T is identical for all modes, implying that the change-in-basis of our state vector remains consistent over time. Our choice of the number of columns of zeros in C is, in essence, a choice of model order, and one which we address in Sec. 4.1.1.

For the HDP-AR-HMM, we similarly place a prior on the dynamic parameters, which in this case consist of $\{A_1^{(k)}, \dots, A_r^{(k)}, \Sigma^{(k)}\}$.

The specific choices of priors we explore, and the resulting posterior distributions conditioned on a set of observations, are described in Sec. 4.1.1. The Gibbs sampling inference scheme is derived in Sec. 4.1.2, and iterates between the following steps for the HDP-SLDS:

1. Sample the state sequence $\boldsymbol{x}_{1:T}$ given the mode sequence $z_{1:T}$ and SLDS parameters $\{A^{(k)}, \Sigma^{(k)}, R\}$.
2. Sample the mode sequence $z_{1:T}$ given the state sequence $\boldsymbol{x}_{1:T}$, HMM parameters $\{\pi_k\}$, and dynamic parameters $\{A^{(k)}, \Sigma^{(k)}\}$.
3. Sample the HMM parameters $\{\pi_k\}$ and SLDS parameters $\{A^{(k)}, \Sigma^{(k)}, R\}$ given the sequences, $z_{1:T}$, $\boldsymbol{x}_{1:T}$, and $\boldsymbol{y}_{1:T}$.

The sampler for the HDP-AR-HMM reuses many steps of the HDP-SLDS sampler, except for directly sampling the mode sequence $z_{1:T}$ based on the *observations* $\boldsymbol{y}_{1:T}$ (whereas the HDP-SLDS relied on a hidden, sampled state sequence $\boldsymbol{x}_{1:T}$ for this step.) Then, resampling the dynamic parameters simply uses the sequences $z_{1:T}$ and $\boldsymbol{y}_{1:T}$. In order to make the connections between the samplers for the HDP-SLDS and HDP-AR-HMM explicit in the following sections, we introduce the concept of *pseudo-observations* $\boldsymbol{\psi}_{1:T}$. For the HDP-AR-HMM, the pseudo-observations are simply the original observations $\boldsymbol{y}_{1:T}$. However, for the HDP-SLDS, the pseudo-observations are the sampled state sequence $\boldsymbol{x}_{1:T}$. A block diagram depicting the connections between these samplers is shown in Fig. 4.2.

■ 4.1.1 Posterior Inference of Dynamic Parameters

In this section we focus on developing a prior to regularize the learning of the dynamic parameters (and measurement noise) conditioned on a fixed mode assignment $z_{1:T}$. The joint learning of the number of modes and resampling the mode sequence $z_{1:T}$ follows as a straightforward extension of the methods described for the sticky HDP-HMM in Chapter 3.

To analyze the posterior distribution of the dynamic parameters, it is useful to first rewrite the dynamic equation for both the HDP-SLDS and HDP-AR-HMM generically in terms of the pseudo-observations $\boldsymbol{\psi}_t$ in the following manner:

$$\boldsymbol{\psi}_t = \mathbf{A}^{(k)} \bar{\boldsymbol{\psi}}_{t-1} + \boldsymbol{e_t}, \tag{4.5}$$

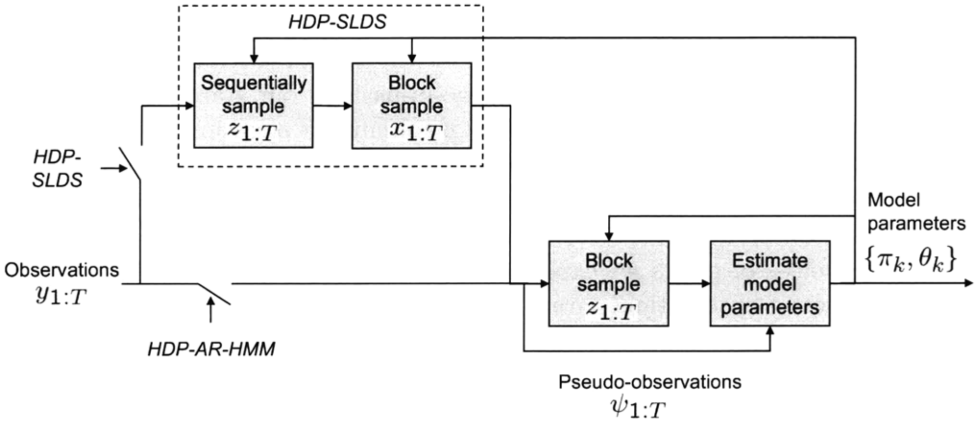

HDP-SLDS
Sequentially sample $z_{1:T}$
Block sample $x_{1:T}$
HDP-SLDS
Observations $y_{1:T}$
HDP-AR-HMM
Model parameters $\{\pi_k, \theta_k\}$
Block sample $z_{1:T}$
Estimate model parameters
Pseudo-observations $\psi_{1:T}$

Figure 4.2. Block diagram of one iteration of the Gibbs sampler for the HDP-SLDS and HDP-AR-HMM. Assume there exists a previous sample of model parameters. Based on the model type, the appropriate switch closes. If HDP-SLDS, there is an extra stage of sampling the latent, continuous state sequence $\boldsymbol{x}_{1:T}$. This sequence then becomes the pseudo-observations. If HDP-AR-HMM, the pseudo-observations are simply the original observations $\boldsymbol{y}_{1:T}$. The pseudo-observations are then used to block sample the mode sequence $z_{1:T}$. Subsequently, a new set of model parameters are sampled conditioned on the mode sequence and pseudo-observations.

where we utilize the following definitions:

	HDP-AR-HMM	HDP-SLDS
Dynamic matrix	$\mathbf{A}^{(k)} = [A_1^{(k)} \dots A_r^{(k)}] \in \mathbb{R}^{d \times (d*r)}$	$\mathbf{A}^{(k)} = A^{(k)} \in \mathbb{R}^{n \times n}$
Pseudo-observations	$\boldsymbol{\psi_t} = \boldsymbol{y}_t$	$\boldsymbol{\psi_t} = \boldsymbol{x}_t$
Lag pseudo-observations	$\bar{\boldsymbol{\psi}}_t = [\boldsymbol{y}_{t-1}^T \cdots \boldsymbol{y}_{t-r}^T]^T$	$\bar{\boldsymbol{\psi}}_t = \boldsymbol{x}_{t-1}$.

(4.6)

For the HDP-AR-HMM, we have simply written the dynamic equation of Eq. (4.1) in matrix form by concatenating the lag matrices into a single matrix $\mathbf{A}^{(k)}$ and forming a *lag observation vector* $\bar{\boldsymbol{\psi}}_t$ comprised of a series of previous observation vectors. For the HDP-SLDS, however, we see that inferences based on pseudo-observations actually rely on a fixed, known state sequence $\boldsymbol{x}_{1:T}$. Methods for resampling this state sequence are discussed in Sec. 4.1.2. For this section (for the HDP-SLDS), we assume such a sample of the state sequence (and hence $\{\boldsymbol{\psi}_t, \bar{\boldsymbol{\psi}}_t\}$) is available so that Eq. (4.5) applies equally well to both the HDP-SLDS and the HDP-AR-HMM.

Conditioned on the mode sequence, one may partition this dynamic sequence into K different linear regression problems, where $K = |\{z_1, \dots, z_T\}|$. That is, for each mode k, we may form a matrix $\boldsymbol{\Psi}^{(k)}$ with N_k columns consisting of the $\boldsymbol{\psi}_t$ with $z_t = k$. Then,

$$\boldsymbol{\Psi}^{(k)} = \mathbf{A}^{(k)} \bar{\boldsymbol{\Psi}}^{(k)} + \mathbf{E}^{(k)}, \tag{4.7}$$

where $\bar{\boldsymbol{\Psi}}^{(k)}$ is a matrix of the associated $\bar{\boldsymbol{\psi}}_{t-1}$, and $\mathbf{E}^{(k)}$ the associated noise vectors.

Conjugate Prior on $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$

The *matrix-normal inverse-Wishart* (MNIW) prior (see Sec. 2.4.4) is conjugate to the likelihood model defined in Eq. (4.7) for the parameter set $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$. Although this prior is typically used for inferring the parameters of a single linear regression problem, it is equally applicable to our scenario since the linear regression problems of Eq. (4.7) are independent conditioned on the mode sequence $z_{1:T}$.

We note that although the MNIW prior does not enforce stability constraints on each mode, this prior is still a reasonable choice since each mode need not have stable dynamics for the SLDS to be stable [30], and conditioned on data from a stable mode, the posterior distribution will likely be sharply peaked around stable dynamic matrices.

Let $\mathbf{D}^{(k)} = \{\mathbf{\Psi}^{(k)}, \bar{\mathbf{\Psi}}^{(k)}\}$. The posterior distribution of the dynamic parameters for the k^{th} mode decomposes as

$$p(\mathbf{A}^{(k)}, \Sigma^{(k)} \mid \mathbf{D}^{(k)}) = p(\mathbf{A}^{(k)} \mid \Sigma^{(k)}, \mathbf{D}^{(k)}) p(\Sigma^{(k)} \mid \mathbf{D}^{(k)}). \tag{4.8}$$

The resulting posterior of $\mathbf{A}^{(k)}$ is derived in Appendix F.1 to be

$$p(\mathbf{A}^{(k)} \mid \Sigma^{(k)}, \mathbf{D}^{(k)}) = \mathcal{MN}\left(\mathbf{A}^{(k)}; \mathbf{S}^{(k)}_{\psi\bar{\psi}} \mathbf{S}^{-(k)}_{\bar{\psi}\bar{\psi}}, \Sigma^{-(k)}, \mathbf{S}^{(k)}_{\bar{\psi}\bar{\psi}}\right), \tag{4.9}$$

with $\boldsymbol{B}^{-(k)}$ denoting $(\boldsymbol{B}^{(k)})^{-1}$ for a given matrix $\boldsymbol{B}$, and

$$\begin{gathered} \mathbf{S}^{(k)}_{\bar{\psi}\bar{\psi}} = \bar{\mathbf{\Psi}}^{(k)} \bar{\mathbf{\Psi}}^{(k)^T} + K \qquad \mathbf{S}^{(k)}_{\psi\bar{\psi}} = \mathbf{\Psi}^{(k)} \bar{\mathbf{\Psi}}^{(k)^T} + MK \\ \mathbf{S}^{(k)}_{\psi\psi} = \mathbf{\Psi}^{(k)} \mathbf{\Psi}^{(k)^T} + MKM^T, \end{gathered} \tag{4.10}$$

where M and K are the parameters defining the matrix-normal portion of the MNIW prior, as in Eq. (2.94).

Assuming an inverse-Wishart prior $\mathrm{IW}(n_0, S_0)$ on $\Sigma^{(k)}$, with S_0 the scale matrix and n_0 the degrees of freedom, Eq. (2.99) gives us the marginal posterior of $\Sigma^{(k)}$:

$$p(\Sigma^{(k)} \mid \mathbf{D}^{(k)}) = \mathrm{IW}\left(N_k + n_0, \mathbf{S}^{(k)}_{\psi|\bar{\psi}} + S_0\right), \tag{4.11}$$

where $\mathbf{S}^{(k)}_{\psi|\bar{\psi}} = \mathbf{S}^{(k)}_{\psi\psi} - \mathbf{S}^{(k)}_{\psi\bar{\psi}} \mathbf{S}^{-(k)}_{\bar{\psi}\bar{\psi}} \mathbf{S}^{(k)^T}_{\psi\bar{\psi}}$ and and $N_k = |\{t \mid z_t = k, t = 1, \ldots, T\}|$.

Alternative Prior — Automatic Relevance Determination

The MNIW prior leads to full $\mathbf{A}^{(k)}$ matrices, which (i) becomes problematic as the model order grows in the presence of limited data; and (ii) does not provide a method for identifying irrelevant model components (i.e. state components in the case of the HDP-SLDS or lag components in the case of the HDP-AR-HMM.) To jointly address these issues, we alternatively consider *automatic relevance determination* (ARD) [9, 112, 124], which encourages driving components of the model parameters to zero if their presence is not supported by the data.

For the HDP-SLDS, we harness the concepts of ARD by placing independent, zero-mean, spherically symmetric Gaussian priors on the columns of the dynamic matrix $\mathbf{A}^{(k)}$:

$$p(\mathbf{A}^{(k)}|\alpha^{(k)}) = \prod_{j=1}^{n} \mathcal{N}\left(\boldsymbol{a}_j^{(k)}; 0, \alpha_j^{-(k)} I_n\right). \tag{4.12}$$

Each precision parameter $\alpha_j^{(k)}$ is given a Gamma(a, b) prior. The zero-mean Gaussian prior penalizes non-zero columns of the dynamic matrix by an amount determined by the precision parameters. Iterative estimation of these hyperparameters $\alpha_j^{(k)}$ and the dynamic matrix $\mathbf{A}^{(k)}$ leads to $\alpha_j^{(k)}$ becoming large for columns whose evidence in the data is insufficient for overcoming the penalty induced by the prior. Having $\alpha_j^{(k)} \to \infty$ drives $\boldsymbol{a}_j^{(k)} \to 0$, implying that the j^{th} state component does not contribute to the dynamics of the k^{th} mode. Thus, examining the set of large $\alpha_j^{(k)}$ provides insight into the order of that mode. Looking at the k^{th} dynamical mode alone, having $\boldsymbol{a}_j^{(k)} = 0$ implies that the realization of *that mode* is not minimal since the associated Hankel matrix H of Eq. (2.154) has reduced rank. However, the overall SLDS realization may still be minimal.

In order to ensure that our choice of $C = [I_d \ 0]$ does not interfere with learning a sparse realization if one exists, we must constrain the class of dynamical phenomenon we may analyze. Imagine a realization of an LDS with

$$\tilde{A} = \begin{bmatrix} 0.8 & 0 \\ 0.2 & 0 \end{bmatrix}, \qquad \tilde{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

Then, the transformation to $C = [1 \ 0]$ using

$$T = \begin{bmatrix} 0.5 & 1 \\ 0.5 & -1 \end{bmatrix}$$

leads to

$$A = T^{-1}\tilde{A}T = \begin{bmatrix} 0.5 & 1 \\ 0.15 & 0.3 \end{bmatrix}$$

So, for this example, fixing $C = [1 \ 0]$ would not lead to learning a sparse dynamical matrix A. Thus, in this chapter we restrict ourselves to ARD modeling of dynamical phenomena that satisfy the following criterion.

Criterion 4.1.1. *If for some realization $\mathcal{R}$ a mode k has $\boldsymbol{a}_j^{(k)} = 0$, then that realization must have $\mathbf{c}_j = 0$, where $\mathbf{c}_j$ is the j^{th} column of C. That is, for all possible realizations, the set of observed state vector components is a subset of those relevant to* all *modes. We assume, without loss of generality, that the states are ordered such that $C = [C_0 \ \ 0]$ (i.e., the observed components are the first components of the state vector.)*

For example, if we have a 3-mode SLDS realization $\mathcal{R}$ with

$$\mathbf{A}^{(1)} = \begin{bmatrix} \boldsymbol{a}_1^{(1)} & \boldsymbol{a}_2^{(1)} & \boldsymbol{a}_3^{(1)} & 0 & 0 \end{bmatrix} \quad \mathbf{A}^{(2)} = \begin{bmatrix} \boldsymbol{a}_1^{(2)} & \boldsymbol{a}_2^{(2)} & 0 & \boldsymbol{a}_4^{(2)} & 0 \end{bmatrix}$$
$$\mathbf{A}^{(3)} = \begin{bmatrix} \boldsymbol{a}_1^{(3)} & \boldsymbol{a}_2^{(3)} & \boldsymbol{a}_3^{(3)} & 0 & \boldsymbol{a}_5^{(3)} \end{bmatrix}, \tag{4.13}$$

then the observation matrix must be of the form $C = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & 0 & 0 & 0 \end{bmatrix}$ to satisfy Criterion 4.1.1.

This criterion is sufficient, though not necessary, for maintaining the sparsity within each $\mathbf{A}^{(k)}$ while still fixing $C = [I_d \; 0]$. That is, given there exists a realization $\mathcal{R}_1$ of our dynamical phenomena that satisfies Criterion 4.1.1, the transformation T to an equivalent realization $\mathcal{R}_2$ with $C = [I_d \; 0]$ will maintain the sparsity structure seen in $\mathcal{R}_1$, which we aim to infer with the ARD prior. The above criterion is reasonable for many applications, as we often have observations only of components of the state vector that are essential to *all* modes, but *some* modes may have additional components that affect the dynamics, but are not directly observed. If there does not exist a realization $\mathcal{R}$ satisfying Criterion 4.1.1, we may instead consider a more general model where the measurement equation is mode-specific and we place a prior on $C^{(k)}$ instead of fixing this matrix. However, this model leads to identifiability issues that are considerably less pronounced in the above case.

The ARD prior may also be used to learn variable-order switching VAR processes. Here, the goal is to "turn off" entire *lag blocks* $A_i^{(k)}$ (whereas in the HDP-SLDS we were interested in eliminating columns of the dynamic matrix.) Instead of placing independent Gaussian priors on each column of $\mathbf{A}^{(k)}$ as we did in Eq. (4.12), we decompose the prior over the lag blocks $A_i^{(k)}$:

$$p(\mathbf{A}^{(k)} | \boldsymbol{\alpha}^{(k)}) = \prod_{i=1}^{r} \mathcal{N}\left(\operatorname{vec}(A_i^{(k)}); 0, \alpha_i^{-(k)} I_{d^2}\right). \tag{4.14}$$

Since each element of a given lag block $A_i^{(k)}$ is distributed according to the same precision parameter $\alpha_i^{(k)}$, if that parameter becomes large the entire lag block will tend to zero.

Example 4.1.1. *For an order three HDP-AR-HMM with observations $\boldsymbol{y}_t \in \mathbb{R}^2$, the dynamic matrix is of the form*

$$\mathbf{A}^{(k)} = \left[\begin{bmatrix} | & | \\ \mathbf{a}_1^{(k)} & \mathbf{a}_2^{(k)} \\ | & | \end{bmatrix} \begin{bmatrix} | & | \\ \mathbf{a}_3^{(k)} & \mathbf{a}_4^{(k)} \\ | & | \end{bmatrix} \begin{bmatrix} | & | \\ \mathbf{a}_5^{(k)} & \mathbf{a}_6^{(k)} \\ | & | \end{bmatrix} \right]$$

and is distributed as

$$p(\mathbf{A}^{(k)}|\boldsymbol{\alpha}^{(k)}) = \prod_{i=1}^{r} \mathcal{N}\left(\begin{bmatrix} a_{1,2i-1}^{(k)} \\ a_{2,2i-1}^{(k)} \\ a_{1,2i}^{(k)} \\ a_{2,2i}^{(k)} \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha_i^{-(k)} & 0 & 0 & 0 \\ 0 & \alpha_i^{-(k)} & 0 & 0 \\ 0 & 0 & \alpha_i^{-(k)} & 0 \\ 0 & 0 & 0 & \alpha_i^{-(k)} \end{bmatrix}\right).$$

In order to examine the posterior distribution on the dynamic matrix $\mathbf{A}^{(k)}$, it is useful to consider the Gaussian induced by Eq. (4.12) and Eq. (4.14) on a vectorization of $\mathbf{A}^{(k)}$. We first provide two examples of this transformation, and then a general form followed by a derivation of the posterior distribution.

Example 4.1.2. *Let us consider an HDP-SLDS in which $\boldsymbol{x}_t \in \mathbb{R}^3$ such that*

$$\mathbf{A}^{(k)} = \begin{bmatrix} | & | & | \\ \mathbf{a}_1^{(k)} & \mathbf{a}_2^{(k)} & \mathbf{a}_3^{(k)} \\ | & | & | \end{bmatrix}.$$

Then, the distribution induced on a vectorization of $\mathbf{A}^{(k)}$ is given by:

$$p\left(\begin{bmatrix} \mathbf{a}_1^{(k)T} & \mathbf{a}_2^{(k)T} & \mathbf{a}_3^{(k)T} \end{bmatrix}^T \mid \boldsymbol{\alpha}^{(k)}\right)$$

$$= \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha_1^{-(k)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1^{-(k)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1^{-(k)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_2^{-(k)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_2^{-(k)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_2^{-(k)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_3^{-(k)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_3^{-(k)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_3^{-(k)} \end{bmatrix}\right)$$

Returning to the HDP-AR-HMM example of Example 4.1.1, the induced distribution *is as follows:*

$$
p\left(\left[\mathbf{a}_1^{(k)T} \;\; \mathbf{a}_2^{(k)T} \;\; \dots \;\; \mathbf{a}_5^{(k)T} \;\; \mathbf{a}_6^{(k)T}\right]^T \mid \boldsymbol{\alpha}^{(k)}\right)
$$

$$
= \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix}
\alpha_1^{-(k)} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & \alpha_1^{-(k)} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha_1^{-(k)} & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha_1^{-(k)} & \dots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \dots & \alpha_3^{-(k)} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & \alpha_3^{-(k)} & 0 & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & \alpha_3^{-(k)} & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \alpha_3^{-(k)}
\end{bmatrix} \right)
$$

More generally, our ARD prior on $\mathbf{A}^{(k)}$ is equivalent to a $\mathcal{N}(0, \Sigma_0^{(k)})$ prior on $\text{vec}(\mathbf{A}^{(k)})$, where

$$
\Sigma_0^{(k)} = \text{diag}\left(\alpha_1^{(k)}, \dots, \alpha_1^{(k)}, \dots, \alpha_m^{(k)}, \dots, \alpha_m^{(k)}\right)^{-1}. \tag{4.15}
$$

Here, $m = n$ for the HDP-SLDS with n replicates of each $\alpha_i^{(k)}$, and $m = r$ for the HDP-AR-HMM with d^2 replicates of $\alpha_i^{(k)}$. (Recall that n is the dimension of the HDP-SLDS state vector $\boldsymbol{x}_t$, r the autoregressive order of the HDP-AR-HMM, and d the dimension of the observations $\boldsymbol{y}_t$.) To examine the posterior distribution of $\mathbf{A}^{(k)}$, we note that we may rewrite the state equation as,

$$
\begin{aligned}
\boldsymbol{\psi}_{t+1} &= \left[\; \bar{\boldsymbol{\psi}}_{t,1} I_\ell \;\; \bar{\boldsymbol{\psi}}_{t,2} I_\ell \;\; \cdots \;\; \bar{\boldsymbol{\psi}}_{t,\ell * r} I_\ell \;\right] \text{vec}(\mathbf{A}^{(k)}) + \boldsymbol{e}_{t+1}(k) \qquad \forall t | z_t = k \\
&\triangleq \tilde{\Psi}_t \text{vec}(\mathbf{A}^{(k)}) + \boldsymbol{e}_{t+1}(k),
\end{aligned} \tag{4.16}
$$

where $\ell = n$ for the HDP-SLDS and $\ell = d$ for the HDP-AR-HMM. Using Eq. (4.16), in Appendix F.2 we derive the posterior distribution as

$$
\begin{aligned}
&p(\text{vec}(\mathbf{A}^{(k)}) \mid \mathbf{D}^{(k)}, \Sigma^{(k)}, \boldsymbol{\alpha}^{(k)}) \\
&\qquad = \mathcal{N}^{-1}\left(\sum_{t | z_t = k} \tilde{\Psi}_{t-1}^T \Sigma^{-(k)} \boldsymbol{\psi}_t, \Sigma_0^{-(k)} + \sum_{t | z_t = k} \tilde{\Psi}_{t-1}^T \Sigma^{-(k)} \tilde{\Psi}_{t-1} \right).
\end{aligned} \tag{4.17}
$$

Here, $\mathcal{N}^{-1}(\vartheta, \Lambda)$ represents a Gaussian $\mathcal{N}(\mu, \Sigma)$ with information parameters $\vartheta = \Sigma^{-1}\mu$ and $\Lambda = \Sigma^{-1}$. Given $\mathbf{A}^{(k)}$, and recalling that each precision parameter is gamma distributed, the posterior of $\alpha_\ell^{(k)}$ is given by

$$p(\alpha_\ell^{(k)} \mid \mathbf{A}^{(k)}) = \text{Gamma}\left(a + \frac{|\mathcal{S}_\ell|}{2}, b + \frac{\sum_{(i,j)\in\mathcal{S}_\ell} a_{ij}^{(k)^2}}{2} \right). \tag{4.18}$$

The set $\mathcal{S}_\ell$ contains the indices for which $a_{ij}^{(k)}$ has prior precision $\alpha_\ell^{(k)}$, as illustrated in the following example.

Example 4.1.3. *Returning to the models of Examples 4.1.1- 4.1.2, for the HDP-SLDS the sets $\mathcal{S}_\ell$ are defined as*

$$\mathcal{S}_1 = \{a_{11}^{(k)}, a_{21}^{(k)}, a_{31}^{(k)}\}, \quad \mathcal{S}_2 = \{a_{12}^{(k)}, a_{22}^{(k)}, a_{32}^{(k)}\}, \quad \mathcal{S}_3 = \{a_{13}^{(k)}, a_{32}^{(k)}, a_{33}^{(k)}\}.$$

For the HDP-AR-HMM, we have

$$\mathcal{S}_1 = \{a_{11}^{(k)}, a_{21}^{(k)}, a_{12}^{(k)}, a_{22}^{(k)}\}, \quad \mathcal{S}_2 = \{a_{13}^{(k)}, a_{23}^{(k)}, a_{14}^{(k)}, a_{24}^{(k)}\}, \quad \mathcal{S}_3 = \{a_{15}^{(k)}, a_{25}^{(k)}, a_{16}^{(k)}, a_{26}^{(k)}\}.$$

Note that in this model, regardless of the number of observations $\boldsymbol{y}_t$, the size of $\mathcal{S}_\ell$ (i.e., the number of $a_{ij}^{(k)}$ used to inform the posterior distribution) remains the same. Thus, the gamma prior is an informative prior and the choice of a and b should depend upon the cardinality of $\mathcal{S}_\ell$. For the HDP-SLDS, this cardinality is given by the maximal state dimension n, and for the HDP-AR-HMM, by the square of the observation dimensionality d^2.

We then place a separate inverse-Wishart prior $\text{IW}(n_0, S_0)$ on $\Sigma^{(k)}$ and look at the posterior given $\mathbf{A}^{(k)}$:

$$p(\Sigma^{(k)} \mid \mathbf{D}^{(k)}, \mathbf{A}^{(k)}) = \text{IW}\left(N_k + n_0, \mathbf{S}_{\psi|\bar{\psi}}^{(k)} + S_0 \right), \tag{4.19}$$

where here, as opposed to in Eq. (4.11), we define

$$\mathbf{S}_{\psi|\bar{\psi}}^{(k)} = \sum_{t|z_t=k} (\boldsymbol{\psi}_t - \mathbf{A}^{(k)}\bar{\boldsymbol{\psi}}_{t-1})(\boldsymbol{\psi}_t - \mathbf{A}^{(k)}\bar{\boldsymbol{\psi}}_{t-1})^T. \tag{4.20}$$

Measurement Noise Posterior

For the HDP-SLDS, we additionally place an $\text{IW}(r_0, R_0)$ prior on the measurement noise covariance R, which we assume is shared between modes. The posterior distribution is given by

$$p(R \mid \boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}) = \text{IW}(T + r_0, S_R + R_0), \tag{4.21}$$

where $S_R = \sum_{t=1}^T (\boldsymbol{y}_t - C\boldsymbol{x}_t)(\boldsymbol{y}_t - C\boldsymbol{x}_t)^T$. If we wished to consider a model with mode-specific measurement noise, we would simply partition the data according to the mode sequence and examine the posterior:

$$p(R^{(k)} \mid \boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, z_{1:T}) = \text{IW}\left(N_k + r_0, S_R^{(k)} + R_0 \right) \quad \forall\, k, \tag{4.22}$$

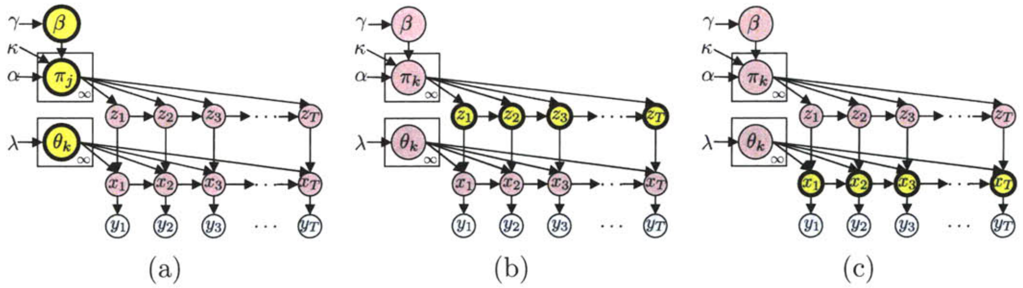


Figure 4.3. Depiction of the HDP-SLDS sampling stages. (a) Sampling of transition and dynamic parameters β, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}$ conditioned on a sampled state sequence $\boldsymbol{x}_{1:T}$ and mode sequence $z_{1:T}$. (b) Block sampling of the mode sequence $z_{1:T}$ conditioned on the sampled state sequence $\boldsymbol{x}_{1:T}$, transition distributions $\boldsymbol{\pi}$, and dynamic parameters $\boldsymbol{\theta}$. (c) Block sampling of state sequence $\boldsymbol{x}_{1:T}$ conditioned on the sampled mode sequence $z_{1:T}$, dynamic parameters $\boldsymbol{\theta}$, and observations $\boldsymbol{y}_{1:T}$. Hyperparameters κ,α, and γ are additionally sampled, but omitted here for simplicity. This animation also ignores the optional stage of sequentially sampling $z_{1:T}$ marginalizing $\boldsymbol{x}_{1:T}$.

where $S_R^{(k)} = \sum_{t|z_t=k}(\boldsymbol{y}_t - C\boldsymbol{x}_t)(\boldsymbol{y}_t - C\boldsymbol{x}_t)^T$. The derivations in the appendices consider this more general case, while the subsequent sections of this chapter maintain the assumption that the measurement mechanism is independent of the dynamical regime.

■ 4.1.2 Gibbs Sampler

For inference in the HDP-AR-HMM, we use a Gibbs sampler that iterates between sampling the mode sequence, $z_{1:T}$, and the set of dynamic and sticky HDP-HMM parameters. The sampler for the HDP-SLDS is identical with the additional step of sampling the state sequence, $\boldsymbol{x}_{1:T}$, and conditioning on this sequence when resampling dynamic parameters and the mode sequence. Periodically, we interleave a step that sequentially samples the mode sequence $z_{1:T}$ marginalizing over the state sequence $\boldsymbol{x}_{1:T}$ in a similar vain to that of Carter and Kohn [26]. We describe the sampler in terms of the pseudo-observations $\boldsymbol{\psi}_t$, as defined by Eq. (4.5), in order to clearly specify the sections of the sampler shared by both the HDP-AR-HMM and HDP-SLDS. Refer to Fig. 4.2 and Figs. 4.3-4.4.

Sampling Dynamic Parameters $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$

Conditioned on the mode sequence, $z_{1:T}$, and the pseudo-observations, $\boldsymbol{\psi}_{1:T}$, we can sample the dynamic parameters $\boldsymbol{\theta} = \{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$ from the posterior densities of Sec. 4.1.1. For the ARD prior, we then sample $\boldsymbol{\alpha}^{(k)}$ given $\mathbf{A}^{(k)}$. In practice we iterate multiple times between sampling $\boldsymbol{\alpha}^{(k)}$ given $\mathbf{A}^{(k)}$ and $\mathbf{A}^{(k)}$ given $\boldsymbol{\alpha}^{(k)}$ before moving to the next sampling stage.

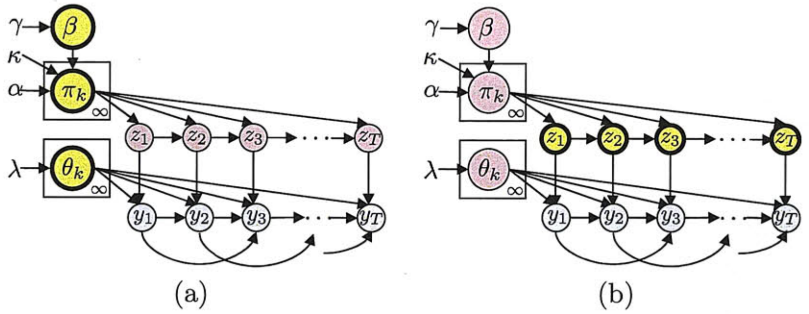


Figure 4.4. Depiction of the HDP-AR-HMM sampling stages. (a) Sampling of transition and dynamic parameters β, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}$ conditioned on the observations $\boldsymbol{y}_{1:T}$ and a sampled mode sequence $z_{1:T}$. (b) Block sampling of the mode sequence $z_{1:T}$ conditioned on the observations $\boldsymbol{y}_{1:T}$, transition distributions $\boldsymbol{\pi}$, and dynamic parameters $\boldsymbol{\theta}$. Hyperparameters κ,α, and γ are additionally sampled, but omitted here for simplicity.

Sampling Measurement Noise R (HDP-SLDS only)

For the HDP-SLDS, we additionally sample the measurement noise covariance R conditioned on the sampled state sequence $\boldsymbol{x}_{1:T}$. In this case, we use the notation $\boldsymbol{\theta}$ to represent the set of dynamic parameters *and* the measurement noise covariance. Namely, $\boldsymbol{\theta} = \{\mathbf{A}^{(k)}, \Sigma^{(k)}, R\}$.

Block Sampling $z_{1:T}$

As shown in Chapter 3, the mixing rate of the Gibbs sampler for the HDP-HMM can be dramatically improved by using a truncated approximation to the HDP, such as the weak limit approximation, and jointly sampling the mode sequence using a variant of the forward-backward algorithm. In the case of our switching dynamical systems, we must account for the direct correlations in the observations in our likelihood computation. The variant of the forward-backward algorithm we use here then involves computing backward messages $m_{t+1,t}(z_t) \propto p(\boldsymbol{\psi}_{t+1:T}|z_t, \bar{\boldsymbol{\psi}}_t, \boldsymbol{\pi}, \boldsymbol{\theta})$ followed by recursively sampling each z_t conditioned on z_{t-1} from

$$p(z_t \mid z_{t-1}, \boldsymbol{\psi}_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto p(z_t \mid \pi_{z_{t-1}}) p(\boldsymbol{\psi}_t \mid \bar{\boldsymbol{\psi}}_{t-1}, \mathbf{A}^{(z_t)}, \Sigma^{(z_t)}) m_{t+1,t}(z_t). \tag{4.23}$$

Joint sampling of the mode sequence is especially important when the observations are directly correlated via a dynamical process since this correlation further slows the mixing rate of the sampler of Teh et al. [162]. Note that as with the sticky HDP-HMM, using an order L weak limit approximation to the HDP still encourages the use of a sparse subset of the L possible dynamical modes.

Block Sampling $\boldsymbol{x}_{1:T}$ (HDP-SLDS only)

Conditioned on the mode sequence $z_{1:T}$ and the set of SLDS parameters $\boldsymbol{\theta}$, our dynamical process simplifies to a time-varying linear dynamical system. We can then block sample $\boldsymbol{x}_{1:T}$ by first running a backward Kalman filter to compute $m_{t+1,t}(\boldsymbol{x}_t) \propto p(\boldsymbol{y}_{t+1:T}|\boldsymbol{x}_t, z_{t+1:T}, \boldsymbol{\theta})$ and then recursively sampling each $\boldsymbol{x}_t$ conditioned on $\boldsymbol{x}_{t-1}$ from

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) \propto p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, A^{(z_t)}, \Sigma^{(z_t)}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, R) m_{t+1,t}(\boldsymbol{x}_t). \tag{4.24}$$

The messages are given in information form by $m_{t,t-1}(\boldsymbol{x}_{t-1}) \propto \mathcal{N}^{-1}(\boldsymbol{x}_{t-1}; \vartheta_{t,t-1}, \Lambda_{t,t-1})$, where the information parameters are recursively defined as

$$\begin{aligned} \vartheta_{t,t-1} &= A^{(z_t)^T} \Sigma^{-(z_t)} \tilde{\Lambda}_t (C^T R^{-1} \boldsymbol{y}_t + \vartheta_{t+1,t}) \\ \Lambda_{t,t-1} &= A^{(z_t)^T} \Sigma^{-(z_t)} A^{(z_t)} - A^{(z_t)^T} \Sigma^{-(z_t)} \tilde{\Lambda}_t \Sigma^{-(z_t)} A^{(z_t)}, \end{aligned} \tag{4.25}$$

with $\tilde{\Lambda}_t = (\Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t})^{-1}$. See Appendix D for a derivation and for a more numerically stable version of this recursion.

Sequentially Sampling $z_{1:T}$ (HDP-SLDS only)

For the HDP-SLDS, iterating between the previous sampling stages can lead to slow mixing rates since the mode sequence is sampled conditioned on a sample of the state sequence. For high-dimensional state spaces $\mathbb{R}^n$, this problem is exacerbated. Instead, one can analytically marginalize the state sequence and sequentially sample the mode sequence from $p(z_t \mid z_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta})$.[4] This marginalization is accomplished by once again harnessing the fact that conditioned on the mode sequence, our model reduces to a time-varying linear dynamical system. When sampling z_t and conditioning on the mode sequence at all *other* time steps, we can run a forward Kalman filter to marginalize the state sequence $\boldsymbol{x}_{1:t-2}$ producing $p(\boldsymbol{x}_{t-1} \mid y_{1:t-1}, z_{1:t-1}, \boldsymbol{\theta})$, and a backward filter to marginalize $\boldsymbol{x}_{t+1:T}$ producing $p(\boldsymbol{y}_{t+1:T} \mid x_t, z_{t+1:T}, \boldsymbol{\theta})$. Then, for each possible value of z_t, we combine these forward and backward messages with the local likelihood $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$ and local dynamic $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{\theta}, z_t = k)$ and marginalize over $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ resulting in the likelihood of the observation sequence $\boldsymbol{y}_{1:T}$ as a function of $\boldsymbol{z}_t$. This likelihood is combined with the prior probability of transitioning from z_{t-1} to $z_t = k$ and from $z_t = k$ to z_{t+1}. The resulting distribution is given by (see Appendix D.3 for full derivations):

$$\begin{aligned} & p(z_t = k \mid z_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto \pi_{z_{t-1}}(k) \pi_k(z_{t+1}) \\ & \frac{|\Lambda_t^{(k)}|^{1/2}}{|\Lambda_t^{(k)} + \Lambda_{t|t}^b|^{1/2}} \exp\left(-\frac{1}{2} \vartheta_t^{(k)^T} \Lambda_t^{-(k)} \vartheta_t^{(k)} + \frac{1}{2} (\vartheta_t^{(k)} + \vartheta_{t|t}^b)^T (\Lambda_t^{(k)} + \Lambda_{t|t}^b)^{-1} (\vartheta_t^{(k)} + \vartheta_{t|t}^b) \right) \end{aligned} \tag{4.26}$$

with

$$\begin{aligned} \Lambda_t^{(k)} &= (\Sigma^{(k)} + \mathbf{A}^{(z_t)} \Lambda_{t-1|t-1}^{-f} \mathbf{A}^{(z_t)^T})^{-1} \\ \vartheta_t^{(k)} &= (\Sigma^{(k)} + \mathbf{A}^{(z_t)} \Lambda_{t-1|t-1}^{-f} \mathbf{A}^{(z_t)^T})^{-1} \mathbf{A}^{(z_t)} \Lambda_{t-1|t-1}^{-f} \vartheta_{t-1|t-1}^f. \end{aligned} \tag{4.27}$$

[4] Note that the instantiated parameter values are used in this sequential sampling, and the resampling of these parameters relies on the sampled state sequence so that one cannot simply iterate between sampling the mode sequence and parameters.

Here, $\vartheta^f_{t|t}$ and $\Lambda^f_{t|t}$ are the updated information parameters for a forward running Kalman filter, defined recursively as

$$\begin{aligned}\Lambda^f_{t|t} &= C^T R^{-1} C + \Sigma^{-(z_t)} - \Sigma^{-(z_t)} \mathbf{A}^{(z_t)} (\mathbf{A}^{(z_t)^T} \Sigma^{-(z_t)} \mathbf{A}^{(z_t)} + \Lambda^f_{t-1|t-1})^{-1} \mathbf{A}^{(z_t)^T} \Sigma^{-(z_t)} \\ \vartheta^f_{t|t} &= C^T R^{-1} y_t + \Sigma^{-(z_t)} \mathbf{A}^{(z_t)} (\mathbf{A}^{(z_t)^T} \Sigma^{-(z_t)} \mathbf{A}^{(z_t)} + \Lambda^f_{t-1|t-1})^{-1} \vartheta^f_{t-1|t-1}. \end{aligned} \tag{4.28}$$

Note that a sequential node ordering for this sampling step allows for efficient updates to the recursively defined filter parameters. However, this sequential sampling is still computationally intensive, so our Gibbs sampler iterates between blocked sampling of the state and mode sequences many times before interleaving a sequential mode sequence sampling step. The resulting Gibbs sampler is outlined in Algorithm 13, with specifications for the MNIW and ARD priors provided in Algorithm 15 and Algorithm 16, respectively.

■ 4.2 Results

■ 4.2.1 MNIW prior

We begin by analyzing the relative modeling power of the HDP-VAR(1)-HMM[5], HDP-VAR(2)-HMM, and HDP-SLDS using the MNIW prior on three sets of test data displayed in Fig. 4.5. We compare to a baseline sticky HDP-HMM using first difference observations, imitating a HDP-VAR(1)-HMM with $A^{(k)} = I$ for all k. In Fig. 4.6 we display Hamming distance errors that are calculated by choosing the optimal mapping of indices maximizing overlap between the true and estimated mode sequences.

For all of the scenarios, we set the MNIW hyperparameters from statistics of the data in the following way. We start by assuming the mean matrix M is $\mathbf{0}$, and setting $K = I_m$. This choice centers the mass of the matrix-normal distribution around stable dynamic matrices while allowing for considerable variability in the matrix values. The inverse-Wishart portion of the prior is given $n_0 = m + 2$ degrees of freedom, which is the smallest integer setting[6] that maintains a proper prior. Recall that smaller degrees of freedom implies a broader prior distribution.

For the HDP-AR-HMM, the scale matrix S_0 is set to 0.75 times the empirical covariance of the entire dataset. Setting the prior directly from the data can help move the mass of the distribution to reasonable values of the parameter space. Ideally, one would like to account for the uncertainty in $\mathbf{A}^{(k)}$ when setting the distribution of $\Sigma^{(k)}$. However, this presents a challenge since the mode-specific covariance $\Sigma^{(k)}$ factors into the the matrix-normal prior on $\mathbf{A}^{(k)}$. Simply analyzing the observations $\boldsymbol{y}_t$ as if $\mathbf{A}^{(k)} = 0$ for all k is made as a practical choice; our use of small degrees of freedom aids in mitigating the affects of this ad-hoc assumption. The fact that the covariance computed from a pooling of all of the data overestimates the mode-specific covariance motivates our slight downscaling (by a factor of 0.75) of the estimate.

[5] Here we use the notation HDP-VAR(r)-HMM to refer to a HDP-AR-HMM with autoregressive order r and vector observations.

[6] Note that it is not required to set the degrees of freedom to an integer.

```
Given a previous set of mode-specific transition probabilities $\boldsymbol{\pi}^{(n-1)}$, the global transition distribution $\beta^{(n-1)}$, and dynamic parameters $\boldsymbol{\theta}^{(n-1)}$:
1. Set $\boldsymbol{\pi} = \boldsymbol{\pi}^{(n-1)}$, $\boldsymbol{\beta} = \boldsymbol{\beta}^{(n-1)}$, and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n-1)}$.
2. If HDP-SLDS,
   (a) For each $t \in \{1, \dots, T\}$, compute $\{\vartheta^f_{t|t}, \Lambda^f_{t|t}\}$ as in Algorithm 20.
   (b) For each $t \in \{T, \dots, 1\}$,
       i. Compute $\{\vartheta^b_{t|t}, \Lambda^b_{t|t}\}$ as in Algorithm 19
       ii. For each $k \in \{1, \dots, K\}$, compute $\{\vartheta^{(k)}_t, \Lambda^{(k)}_t\}$ as in Eq. (4.27) and set
           $f_k(\boldsymbol{y}_{1:T}) = |\Lambda^{(k)}_t|^{1/2} |\Lambda^{(k)}_t + \Lambda^b_{t|t}|^{-1/2}$
           $\exp\left(-\frac{1}{2}\vartheta^{(k)^T}_t \Lambda^{-(k)}_t \vartheta^{(k)}_t + \frac{1}{2}(\vartheta^{(k)}_t + \vartheta^b_{t|t})^T (\Lambda^{(k)}_t + \Lambda^b_{t|t})^{-1} (\vartheta^{(k)}_t + \vartheta^b_{t|t})\right).$
       iii. Sample a mode assignment
           $z_t \sim \sum_{k=1}^{L} \pi_{z_{t-1}}(k) \pi_k(z_{t+1}) f_k(\boldsymbol{y}_{1:T}) \delta(z_t, k).$
   (c) Working sequentially forward in time sample
       $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_t; (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1}(\Sigma^{-(z_t)} A^{(z_t)} \boldsymbol{x}_{t-1} + \vartheta^b_{t|t}), (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1}).$
   (d) Set pseudo-observations $\boldsymbol{\psi}_{1:T} = \boldsymbol{x}_{1:T}$.
3. If HDP-AR-HMM, set pseudo-observations $\boldsymbol{\psi}_{1:T} = \boldsymbol{y}_{1:T}$.
4. Block sample $z_{1:T}$ given transition distributions $\boldsymbol{\pi}$, dynamic parameters $\boldsymbol{\theta}$, and pseudo-observations $\boldsymbol{\psi}_{1:T}$ as in Algorithm 14.
5. Update the global transition distribution $\beta$ (utilizing auxiliary variables $\boldsymbol{m}$, $\boldsymbol{w}$, and $\bar{\boldsymbol{m}}$), mode-specific transition distributions $\pi_k$, and hyperparameters $\alpha$, $\gamma$, and $\kappa$ as in Algorithm 10 of Sec. 3.1.3.
6. For each $k \in \{1, \dots, L\}$, sample dynamic parameters $(\mathbf{A}^{(k)}, \Sigma^{(k)})$ given the pseudo-observations $\boldsymbol{\psi}_{1:T}$ and mode sequence $z_{1:T}$ as in Algorithm 15 for the MNIW prior and Algorithm 16 for the ARD prior.
7. If HDP-SLDS, also sample the measurement noise covariance
       $R \sim \mathrm{IW}\left(T + r_0, \sum_{t=1}^{T} (\boldsymbol{y}_t - C\boldsymbol{x}_t)(\boldsymbol{y}_t - C\boldsymbol{x}_t)^T + R_0\right).$
8. Fix $\boldsymbol{\pi}^{(n)} = \boldsymbol{\pi}$, $\beta^{(n)} = \beta$, and $\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}$.
```

Algorithm 13. HDP-SLDS and HDP-AR-HMM Gibbs sampler.

```
Given a set of mode-specific transition probabilities $\boldsymbol{\pi}$, dynamic parameters $\boldsymbol{\theta}$, and pseudo-observations $\boldsymbol{\psi}_{1:T}$:
  1. Calculate messages $m_{t,t-1}(k)$, initialized to $m_{T+1,T}(k) = 1$, and the sample mode sequence $z_{1:T}$:
     (a) For each $t \in \{T, \dots, 1\}$ and $k \in \{1, \dots, L\}$, compute
           $m_{t,t-1}(k) = \sum_{j=1}^{L} \pi_k(j) \mathcal{N}\left( \boldsymbol{\psi}_t; \sum_{i=1}^{r} A_i^{(j)} \boldsymbol{\psi}_{t-i}, \Sigma^{(j)} \right) m_{t+1,t}(j)$
     (b) Working sequentially forward in time, starting with transitions counts $n_{jk} = 0$:
         i. For each $k \in \{1, \dots, L\}$, compute the probability
              $f_k(\boldsymbol{\psi}_t) = \mathcal{N}\left( \boldsymbol{y}_t; \sum_{i=1}^{r} A_i^{(k)} \boldsymbol{\psi}_{t-i}, \Sigma^{(k)} \right) m_{t+1,t}(k)$
         ii. Sample a mode assignment $z_t$ as follows and increment $n_{z_{t-1}z_t}$:
              $z_t \sim \sum_{k=1}^{L} \pi_{z_{t-1}}(k) f_k(\boldsymbol{\psi}_t) \delta(z_t, k)$
     Note that the likelihoods can be precomputed for each $k \in \{1, \dots, L\}$.
```

Algorithm 14. Blocked mode-sequence sampler for HDP-AR-HMM or HDP-SLDS.

```
Given pseudo-observations $\boldsymbol{\psi}_{1:T}$ and mode sequence $z_{1:T}$, for each $k \in \{1, \dots, K\}$:
  1. Construct $\boldsymbol{\Psi}^{(k)}$ and $\bar{\boldsymbol{\Psi}}^{(k)}$ as in Eq. (4.7).
  2. Compute sufficient statistics using pseudo-observations $\boldsymbol{\psi}_t$ associated with $z_t = k$:
       $\mathbf{S}_{\bar{\psi}\bar{\psi}}^{(k)} = \bar{\boldsymbol{\Psi}}^{(k)} \bar{\boldsymbol{\Psi}}^{(k)^T} + K$
       $\mathbf{S}_{\psi\bar{\psi}}^{(k)} = \boldsymbol{\Psi}^{(k)} \bar{\boldsymbol{\Psi}}^{(k)^T} + MK$
       $\mathbf{S}_{\psi\psi}^{(k)} = \boldsymbol{\Psi}^{(k)} \boldsymbol{\Psi}^{(k)^T} + MKM^T.$
  3. Sample dynamic parameters:
       $\Sigma^{(k)} \sim \mathrm{IW}\left( N_k + n_0, \mathbf{S}_{\psi|\bar{\psi}}^{(k)} + S_0 \right)$
       $\mathbf{A}^{(k)} \mid \Sigma^{(k)} \sim \mathcal{MN}\left( \mathbf{A}^{(k)}; \mathbf{S}_{\psi\bar{\psi}}^{(k)} \mathbf{S}_{\bar{\psi}\bar{\psi}}^{-(k)}, \Sigma^{-(k)}, \mathbf{S}_{\bar{\psi}\bar{\psi}}^{(k)} \right).$
```

Algorithm 15. Parameter sampling using MNIW prior.

Given pseudo-observations $\boldsymbol{\psi}_{1:T}$, mode sequence $z_{1:T}$, and a previous set of dynamic parameters $(\mathbf{A}^{(k)}, \Sigma^{(k)}, \boldsymbol{\alpha}^{(k)})$, for each $k \in \{1, \dots, K\}$:

1. Construct $\tilde{\Psi}_t$ as in Eq. (4.16).

2. Iterate multiple times between the following steps:

   (a) Construct $\Sigma_0^{(k)}$ given $\boldsymbol{\alpha}^{(k)}$ as in Eq. (4.15) and sample the dynamic matrix:

   $$\begin{aligned}&\mathrm{vec}(\mathbf{A}^{(k)}) \mid \Sigma^{(k)}, \boldsymbol{\alpha}^{(k)}\\ &\qquad \sim \mathcal{N}^{-1}\left(\sum_{t \mid z_t = k} \tilde{\Psi}_{t-1}^T \Sigma^{-(k)} \boldsymbol{\psi}_{\boldsymbol{t}}, \Sigma_0^{-(k)} + \sum_{t \mid z_t = k} \tilde{\Psi}_{t-1}^T \Sigma^{-(k)} \tilde{\Psi}_{t-1} \right).\end{aligned}$$

   (b) For each $\ell \in \{1, \dots, m\}$, with $m = n$ for the SLDS and $m = r$ for the switching VAR, sample ARD precision parameters:

   $$\alpha_\ell^{(k)} \mid \mathbf{A}^{(k)} \sim \text{Gamma}\left(a + \frac{|\mathcal{S}_\ell|}{2}, b + \frac{\sum_{(i,j) \in \mathcal{S}_\ell} a_{ij}^{(k)^2}}{2} \right).$$

   (c) Compute sufficient statistic:

   $$\mathbf{S}_{\psi|\bar{\psi}}^{(k)} = \sum_{t \mid z_t = k} (\boldsymbol{\psi}_t - \mathbf{A}^{(k)} \bar{\boldsymbol{\psi}}_{t-1})(\boldsymbol{\psi}_t - \mathbf{A}^{(k)} \bar{\boldsymbol{\psi}}_{t-1})^T$$

   and sample process noise covariance:

   $$\Sigma^{(k)} \mid \mathbf{A}^{(k)} \quad \sim \quad \text{IW}\left(N_k + n_0, \mathbf{S}_{\psi|\bar{\psi}}^{(k)} + S_0 \right).$$

Algorithm 16. Parameter sampling using ARD prior.

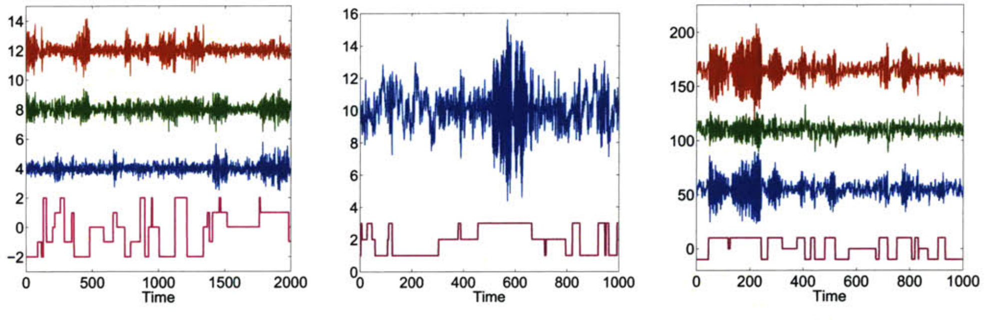
When setting the HDP-SLDS inverse-Wishart prior on $\Sigma^{(k)}$, taking $\mathbf{A}^{(k)} = 0$ still leaves us with ambiguity between the contribution from the measurement and process noise terms in driving the covariance of the observations $\boldsymbol{y}_t$. In addition, for an HDP-SLDS with $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^d$, $n > d$ (i.e., larger state dimension than observation dimension,) we need a method for setting the mean of the extra $n - d$ dimensions of the process noise covariance. Thus, for the HDP-SLDS we use the following heuristic: we set the upper $d \times d$ lefthand quadrant of the scale matrix S_0 to be 0.675 times the empirical covariance; the $n - d \times n - d$ lower righthand quadrant is set to be diagonal with determinant equal to that of the upper righthand $d \times d$ block. We then set the inverse-Wishart prior on the measurement noise to have $r_0 = d + 2$ degrees of freedom and a scale matrix equal to 0.075 times the empirical covariance. Although we have chosen this specific heuristic for setting the hyperparameters of the MNIW prior, we have found that the results are fairly robust to various settings.

As in Chapter 3, we place a Gamma(a, b) prior on the sticky HDP-HMM concentration parameters $\alpha + \kappa$ and γ, and a Beta(c, d) prior on the self-transition proportion parameter $\rho = \kappa/(\alpha + \kappa)$. We once again choose the weakly informative setting of $a = 1$, $b = 0.01$, $c = 10$, and $d = 1$.

For the first scenario (Fig. 4.5(a)), the data were generated from a five-mode switching VAR(1) process with a 0.98 probability of self-transition and equally likely transitions to the other modes. The same mode-transition structure was used in the subsequent two scenarios, as well. The three switching linear dynamical models provide comparable performance since both the HDP-VAR(2)-HMM and HDP-SLDS with $C = I_3$ contain the class of HDP-VAR(1)-HMMs. In the second scenario (Fig. 4.5(b)), the data were generated from a 3-mode switching AR(2) process. The HDP-AR(2)-HMM has significantly better performance than the HDP-AR(1)-HMM while the performance of the HDP-SLDS with $C = [1\ 0]$ performs similarly, but has greater posterior variability because the HDP-AR(2)-HMM model family is smaller. Note that the HDP-SLDS sampler is slower to mix since the hidden, continuous state is also sampled. The data in the third scenario (Fig. 4.5(c)) were generated from a three-mode SLDS model with $C = I_3$. Here, we clearly see that neither the HDP-VAR(1)-HMM nor HDP-VAR(2)-HMM is equivalent to the HDP-SLDS. Note that all of the switching models yielded significant improvements relative to the baseline sticky HDP-HMM. This input representation is more effective than using raw observations for HDP-HMM learning, but still much less effective than richer models which switch among learned LDS. Together, these results demonstrate both the differences between our models as well as the models' ability to learn switching processes with varying numbers of modes.

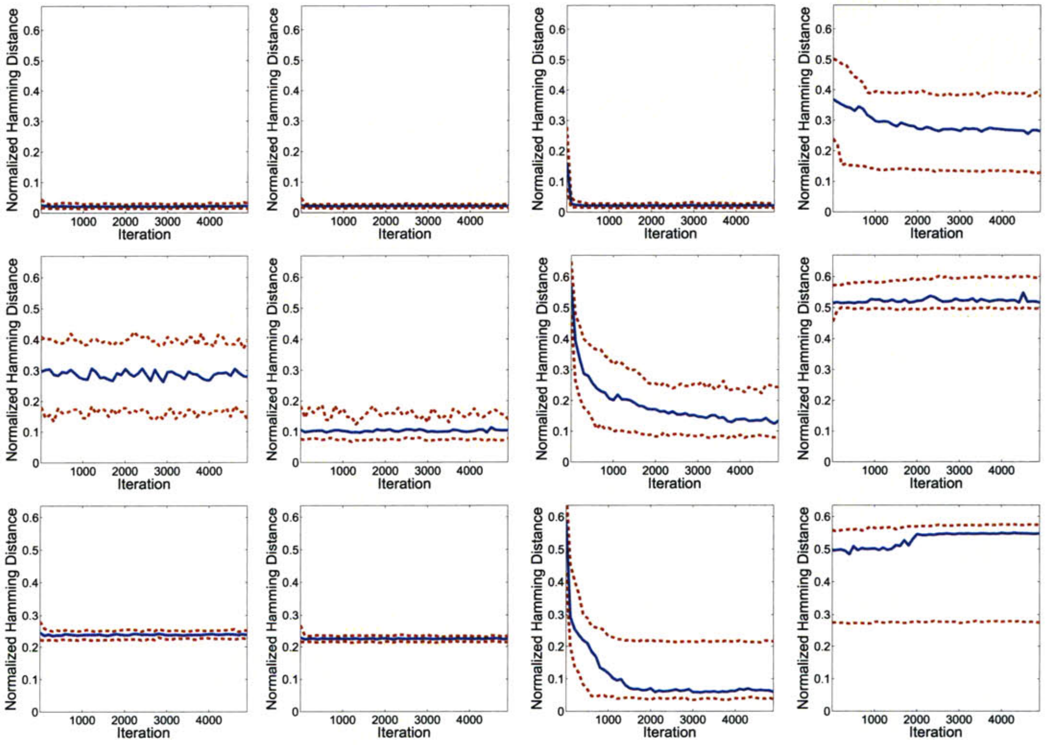
■ 4.2.2 ARD prior

We now compare the utility of the ARD prior to the MNIW prior using the HDP-SLDS model when the true underlying dynamical modes have sparse dependencies relative



(a) (b) (c)

Figure 4.5. (a) Observation sequence (blue, green, red) and associated mode sequence (magenta) for: (a) 5-mode switching VAR(1) process, (b) 3-mode switching AR(2) process, and (c) 3-mode SLDS.



(a) (b) (c) (d)

Figure 4.6. Each row corresponds to an observation sequence of Fig. 4.5: (top) 5-mode switching VAR(1) process, (middle) 3-mode switching AR(2) process, and (bottom) 3-mode SLDS. The associated 10th, 50th, and 90th Hamming distance quantiles over 100 trials are shown for the (b) HDP-VAR(1)-HMM, (c) HDP-VAR(2)-HMM, (d) HDP-SLDS with $C = I$ (top and bottom) and $C = [1 \;\; 0]$ (middle), and (e) sticky HDP-HMM using first difference observations.

to the assumed model order[7]. We generated data from a two-mode SLDS with 0.98 probability of self-transition and the following dynamic parameters:

$$\mathbf{A}^{(1)} = \begin{bmatrix} 0.8 & -0.2 & 0 \\ -0.2 & 0.8 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{A}^{(2)} = \begin{bmatrix} -0.2 & 0 & 0.8 \\ 0.8 & 0 & -0.2 \\ 0 & 0 & 0 \end{bmatrix},$$

with $C = [I_2 \ 0]$, $\Sigma^{(1)} = \Sigma^{(2)} = I_3$, and $R = I_2$. The first dynamical process can be equivalently described by just the first and second state components since the third component is simply white noise that does not contribute to the state dynamics and is not directly (or indirectly) observed. For the second dynamical process, the third state component is once again a white noise process, but *does* contribute to the dynamics of the first and second state components. However, we can equivalently represent the dynamics of this mode as:

$$\begin{aligned} x_{1,t} &= -0.2x_{1,t-1} + \tilde{e}_{1,t} \\ x_{2,t} &= 0.8x_{1,t-1} + \tilde{e}_{2,t} \end{aligned}$$

where $\tilde{\boldsymbol{e}}_t$ is a noise term defined by the original process noise and $x_{3,t}$, and can be shown to be white. We also notice that the second state component solely relies on the first component of the lagged state vector. Thus, one could rewrite the second dynamical mode as a linear dynamical system with

$$\tilde{\mathbf{A}}^{(2)} = \begin{bmatrix} -0.2 & 0 & 0 \\ 0.8 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and process noise covariance appropriately redefined.

Notice that this SLDS does not satisfy Criterion 4.1.1 since the second column of $\mathbf{A}^{(2)}$ is zero while the second column of C is not. That is, the second state component does not contribute to the dynamics of the second mode, but is directly observed. Nevertheless, because the realization is in our canonical form with $C = [I_2 \ 0]$, we still expect to recover the $\boldsymbol{a}_2^{(2)} = \boldsymbol{a}_3^{(2)} = 0$ sparsity structure. We set the parameters of the Gamma(a, b) prior on the ARD precisions as $a = |\mathcal{S}_\ell|$ and $b = a/1000$, where we recall the definition of $\mathcal{S}_\ell$ from Eq. (4.18). This specification fixes the mean of the prior to 1000 while aiming to provide a prior that is equally informative for various choices of model order (i.e., sizes $|\mathcal{S}_\ell|$).

In Fig. 4.7, we see that even in this low-dimensional example, the ARD provides superior mode-sequence estimates, as well as a mechanism for identifying non-dynamical state components. The histograms of the inferred $\boldsymbol{\alpha}^{(k)}$ are shown in Fig. 4.7(d)-(e). From the clear separation between the sampled dynamic range of $\alpha_3^{(1)}$ and $(\alpha_1^{(1)}, \alpha_2^{(1)})$,

[7] That is, the HDP-SLDS may have dynamical regimes reliant on lower state dimensions, or the HDP-AR-HMM may have modes described by lower order VAR processes.

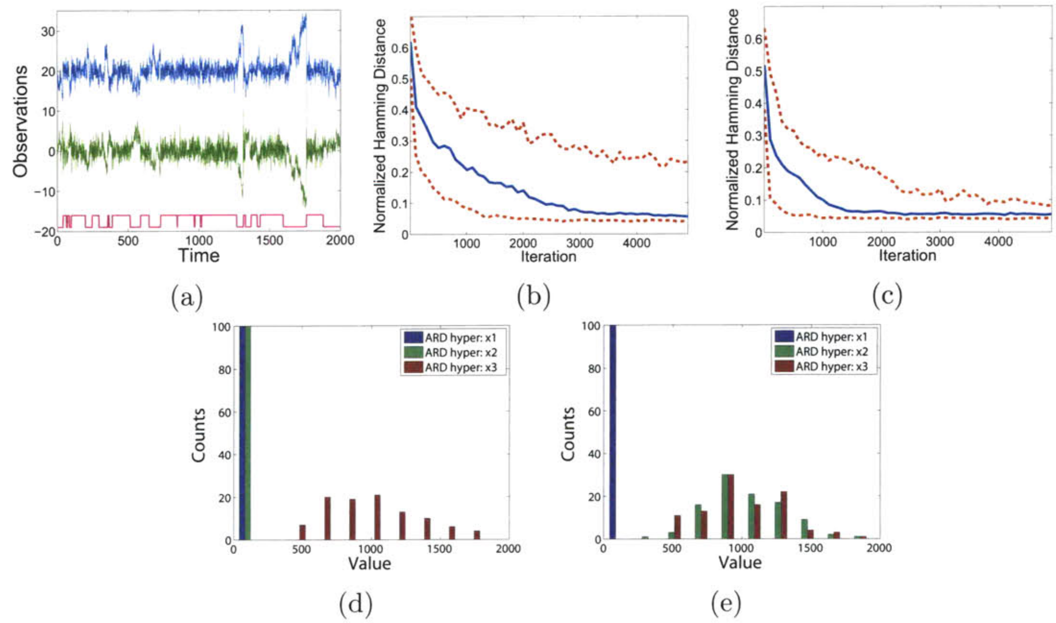


Figure 4.7. (a) Observation sequence (green, blue) and mode sequence (magenta) of a 2-mode SLDS, where the first mode can be realized by the first two state components and the second mode solely by the first. The associated 10th, 50th, and 90th Hamming distance quantiles over 100 trials are shown for the (b) MNIW and (c) ARD prior. (d)-(e) Histograms of inferred ARD precisions associated with the first and second dynamical modes, respectively, at the 5000th Gibbs iteration. Larger values correspond to non-dynamical components.

and between that of $(\alpha_2^{(2)}, \alpha_3^{(2)})$ and $\alpha_1^{(2)}$, we see that we are able to correctly identify dynamical systems with $\boldsymbol{a}_3^{(1)} = 0$ and $\boldsymbol{a}_2^{(2)} = \boldsymbol{a}_3^{(2)} = 0$.

■ 4.2.3 Dancing Honey Bees

Honey bees perform a set of dances within the beehive in order to communicate the location of food sources. Specifically, they switch between a set of *waggle*, *turn-right*, and *turn-left* dances. During the waggle dance, the bee walks roughly in a straight line while rapidly shaking its body from left to right. The turning dances simply involve the bee turning in a clockwise or counterclockwise direction. These turning dances often start at the endpoint of a waggle dance and form a C-like shape that typically returns the bee to the starting location of the waggle dance. We display six such sequences of honey bee dances in Fig. 4.8. The data consist of measurements $\boldsymbol{y}_t = [\cos(\theta_t) \quad \sin(\theta_t) \quad x_t \quad y_t]^T$, where (x_t, y_t) denotes the 2D coordinates of the bee's body and θ_t its head angle[8]. Both Oh et al. [129] and Xuan and Murphy [188] used switching dynamical models to

[8] The data for the six honey bee dance sequences, along with ground truth labels, are available at http://www.cc.gatech.edu/ borg/ijcv_psslds/.

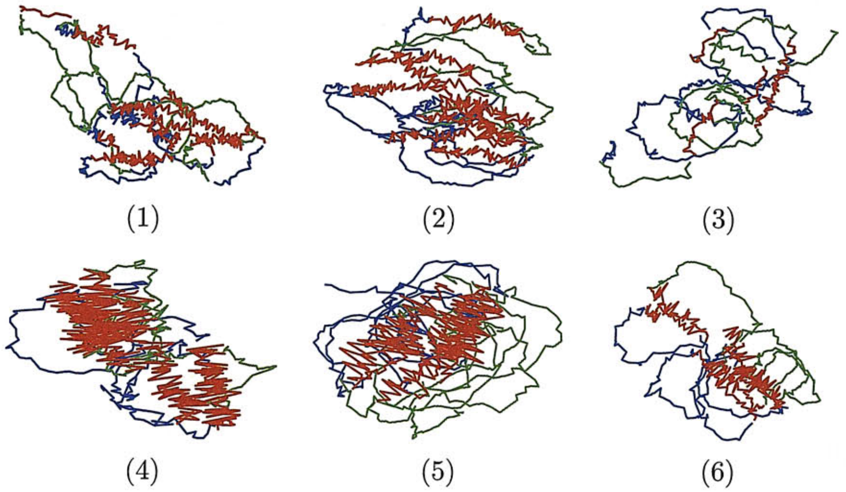


Figure 4.8. Trajectories of the dancing honey bees for sequences 1 to 6, colored by *waggle* (red), *turn right* (blue), and *turn left* (green) dances.

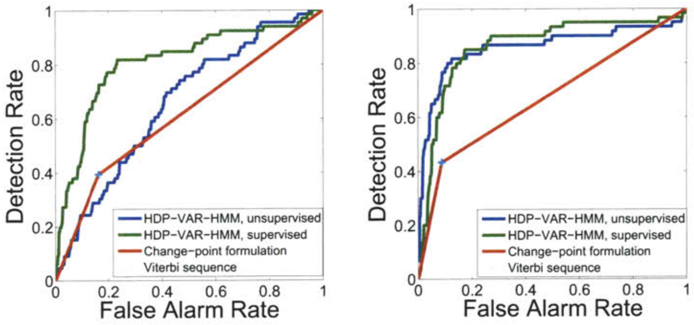
analyze these honey bee dances. We wish to analyze the performance of our Bayesian nonparametric variants of these models in segmenting the six sequences into the dance labels displayed in Fig. 4.8.

MNIW Prior — Unsupervised

We start by testing the HDP-VAR(1)-HMM using a MNIW prior. (Note that we did not see performance gains by considering the HDP-SLDS, so we omit showing results for that architecture.) We set the MNIW prior with mean matrix $M = \mathbf{0}$, $K = 0.1 * I_m$, degrees of freedom $n_0 = 6$, and scale matrix S_0 set to 0.75 times the empirical covariance of a pre-processed observation sequence. The pre-processing involves centering the position observations around 0 and scaling each component of $\boldsymbol{y}_t$ to be within the same dynamic range. As in the synthetic data examples, we set the Gamma(a, b) priors on the concentration parameters $\alpha + \kappa$ and γ to have $a = 1$ and $b = 0.01$, and the Beta(c, d) prior on ρ to have $c = 10$ and $d = 1$.

We compare our results to those of Xuan and Murphy [188], who used a change-point detection technique for inference on this dataset. As shown in Fig. 4.9, our model achieves a superior segmentation compared to the change-point formulation in almost all cases, while also identifying modes which reoccur over time.

Oh et al. [129] also presented an analysis of the honey bee data, using an SLDS with a fixed number of modes. Unfortunately, that analysis is not directly comparable to ours, because Oh et al. [129] used their SLDS in a supervised formulation in which the ground truth labels for all but one of the sequences are employed in the inference of the labels for the remaining held-out sequence, and in which the kernels used in the MCMC procedure depend on the ground truth labels. (The authors also considered a



(a) (b)

Figure 4.9. ROC curves for the unsupervised HDP-VAR-HMM, partially supervised HDP-VAR-HMM, and change-point formulation of Xuan and Murphy [188] using the Viterbi sequence for segmenting datasets (a) 1-3 and (b) 4-6.

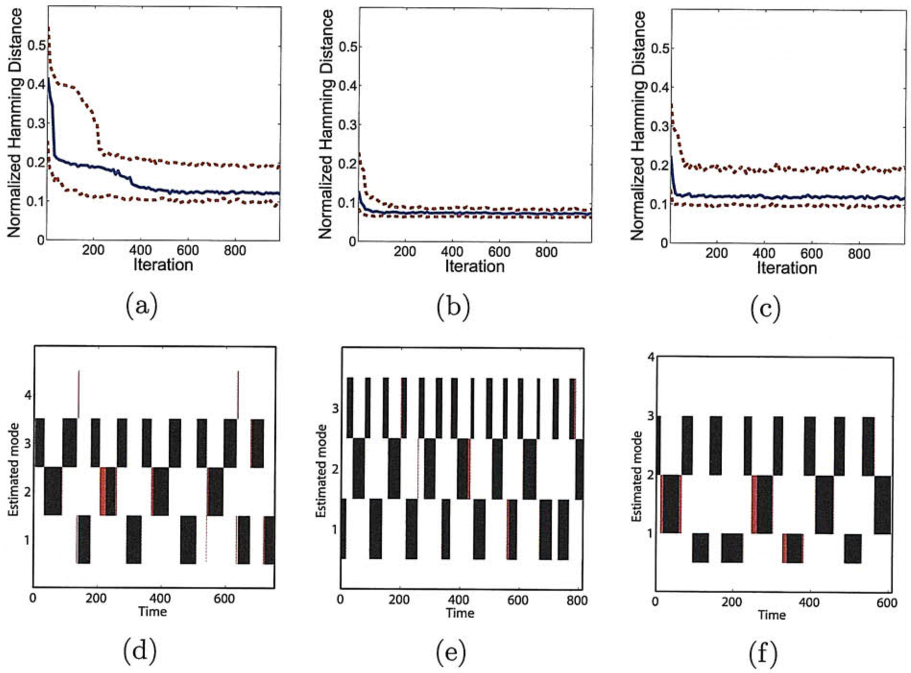
"parameterized segmental SLDS (PS-SLDS)," which makes use of domain knowledge specific to honey bee dancing and requires additional supervision during the learning process.) Nonetheless, in Table 4.1 we report the performance of these methods as well as the median performance (over 100 trials) of the unsupervised HDP-VAR(1)-HMM in order to provide a sense of the level of performance achievable without detailed, manual supervision. As seen in Table 4.1, the HDP-VAR(1)-HMM yields very good performance on sequences 4 to 6 in terms of the learned segmentation and number of modes (see Fig. 4.10); the performance approaches that of the supervised method.

For sequences 1 to 3—which are much less regular than sequences 4 to 6—the performance of the unsupervised procedure is substantially worse. In Fig. 4.11, we see the extreme variation in head angle during the waggle dances of sequences 1 to 3.[9] As noted by Oh, the tracking results based on the vision-based tracker are noisier for these sequences and the patterns of switching between dance modes is more irregular. This dramatically affects our performance since we do not use domain-specific information. Indeed, our learned segmentations consistently identify turn-right and turn-left modes, but often create a new, sequence-specific waggle dance mode. Many of our errors can be attributed to creating multiple waggle dance modes within a sequence. Overall, however, we are able to achieve reasonably good segmentations without having to manually input domain-specific knowledge.

MNIW Prior — Partially Supervised

The discrepancy in performance between our results and the supervised approach of Oh et al. [129] motivated us to also consider a partially supervised variant of the HDP-VAR(1)-HMM in which we fix the ground truth mode sequences for five out of six of

[9] From Fig. 4.11, we also see that even in sequences 4 to 6, the ground truth labeling appear to be inaccurate at times. Specifically, certain time steps are labeled as waggle dances (red) that look more typical of a turning dance (green, blue).

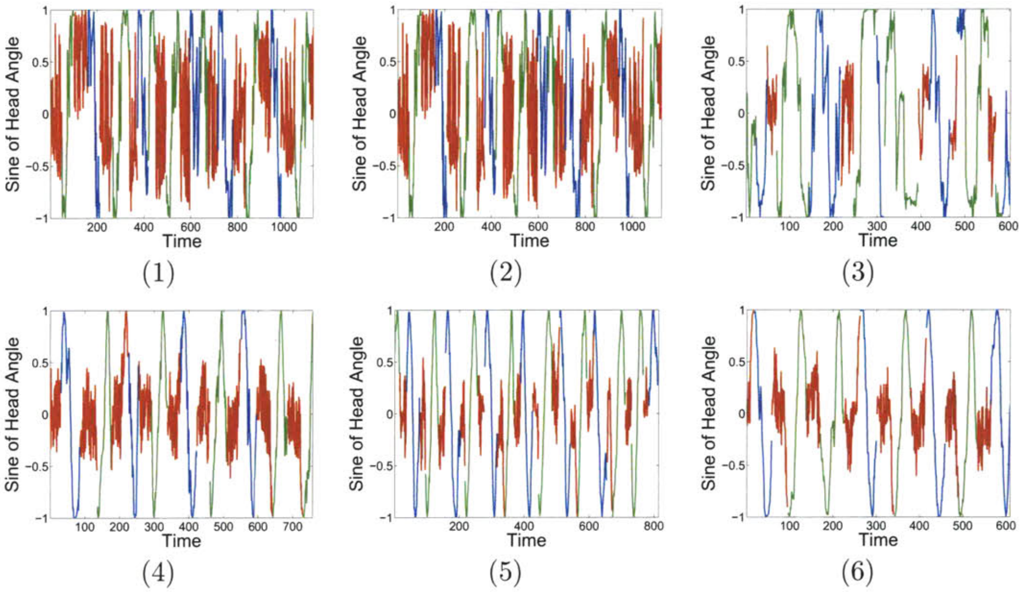


(a) (b) (c)

(d) (e) (f)

Figure 4.10. (a)-(c) The 10th, 50th, and 90th Hamming distance quantiles over 100 trials are shown for sequences 4, 5, and 6, respectively. (d)-(f) Estimated mode sequences representing the median error for sequences 4, 5, and 6 at the 200th Gibbs iteration, with errors indicated in red.

the sequences, and jointly infer both a combined set of dynamic parameters and the left-out mode sequence. This is equivalent to informing the prior distributions with the data from the five fixed sequences, and using these updated posterior distributions as the prior distributions for the held-out sequence. As we see in Table 4.1, this partially supervised approach considerably improved performance for these three sequences, especially sequences 2 and 3. Here, we hand-aligned sequences so that the waggle dances tended to have head angle measurements centered about $\pi/2$ radians. Aligning the waggle dances is possible by looking at the high frequency portions of the head angle measurements. Additionally, the pre-processing of the unsupervised approach is not appropriate here as the scalings and shiftings are dance-specific, and such transformations modify the associated switching VAR(1) model. Instead, to account for the varying frames of reference (i.e., point of origin for each bee body) we allowed for a mean $\boldsymbol{\mu}^{(k)}$ on the process noise, and placed an independent $\mathcal{N}(0, \Sigma_0)$ prior on this parameter. We set Σ_0 to 0.75 times the scale matrix S_0 of the inverse-Wishart prior on $\Sigma^{(k)}$. Since we are not shifting and scaling the observations, we set the scale matrix S_0 to 0.75 times the empirical covariance of the *first difference* observations. We also use $n_0 = 10$ degrees of freedom, making the distribution around the expected covariance tighter than in the unsupervised case. Examining first differences is appropriate since the bee's dynam-



(1) (2) (3) (4) (5) (6)

Figure 4.11. Measurements of the sine of the bee's head angle for the six honey bee dance sequences, colored by the ground truth dance label sequence.

Sequence	1	2	3	4	5	6
HDP-VAR(1)-HMM unsupervised	45.0	42.7	47.3	88.1	92.5	88.2
HDP-VAR(1)-HMM partially supervised	55.0	86.3	81.7	89.0	92.4	89.6
SLDS DD-MCMC	74.0	86.1	81.3	93.4	90.2	90.4
PS-SLDS DD-MCMC	75.9	92.4	83.1	93.4	90.4	91.0

Table 4.1. Median label accuracy of the HDP-VAR(1)-HMM using unsupervised and partially supervised Gibbs sampling, compared to accuracy of the supervised PS-SLDS and SLDS procedures, where the latter algorithms were based on a supervised MCMC procedure (DD-MCMC) [129].

ics are better approximated as a random walk than as i.i.d. observations. Using raw observations in the unsupervised approach creates a larger expected covariance matrix making the prior on the dynamic matrix less informative, which is useful in the absence of other labeled data.

ARD Prior

Using the cleaner sequences 4 to 6, we investigate the honey bee dance's variable order structure by assuming a higher order switching VAR model and employing the ARD prior. Namely, we choose an HDP-VAR(2)-HMM and use the same approach to setting the hyperparameters as in Sec. 4.2.2. Although not depicted here, the Hamming distance plots for the HDP-VAR(2)-HMM with the ARD prior are indistinguishable from

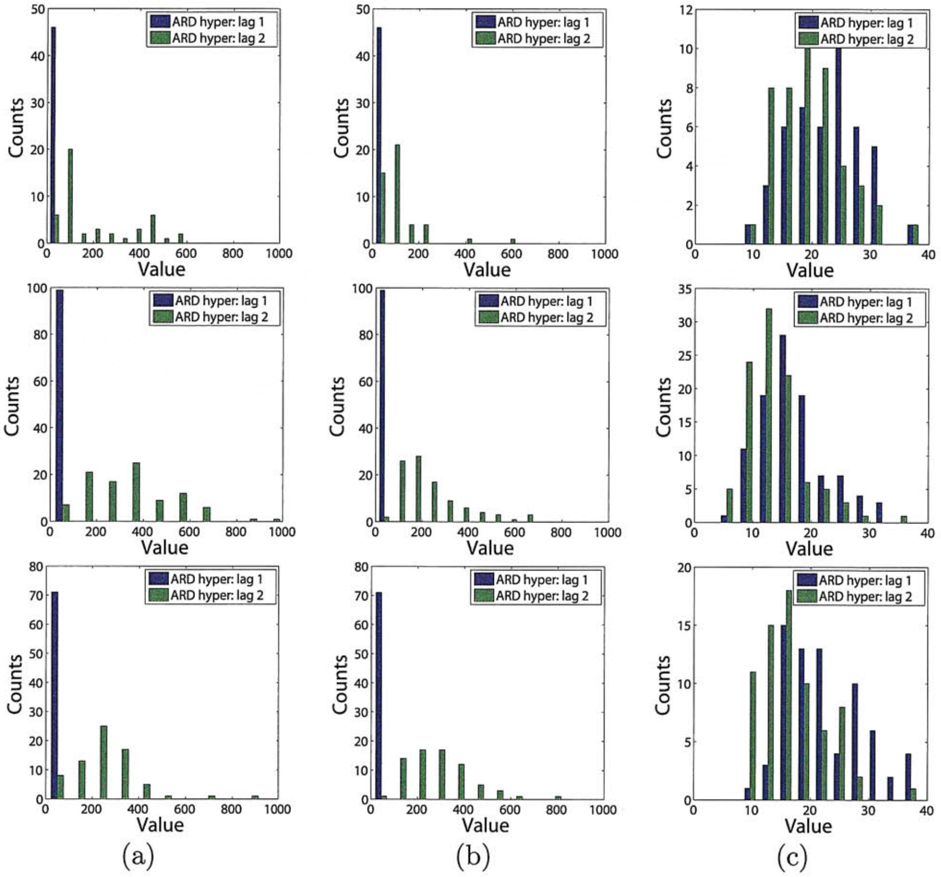


Figure 4.12. (a)-(c) Histograms of the inferred ARD hyperparameters for the learned *turn right*, *turn left*, and *waggle* dance modes, respectively, at the 400th Gibbs iteration for the trials with Hamming distance below the median. Larger values correspond to unnecessary lag components. Note the horizontal axis scale in column (c).

those of Fig. 4.10(a)-(c) using the HDP-VAR(1)-HMM with the MNIW prior. Thus, the information in the first lag component is sufficient for the segmentation problem. However, the ARD prior informs us of the variable-order nature of this switching dynamical process. From Fig. 4.12(a)-(c), we see that the turning dances simply rely on the first lag component while the waggle dance relies on both lag components. To verify these results, we provided the data and ground truth labels to MATLAB's `lpc`[10] implementation of Levinson's algorithm, which indicated that the turning dances are well approximated by an order 1 process, while the waggle dance relies on an order 2 model. Thus, our learned orders for the three dances match what is indicated by Levinson's algorithm on ground-truth segmented data.

[10] `lpc` computes AR coefficients for scalar data, so we analyzed each component of the observation vector independently. The order was consistent across these components.

4.3 Model Variants

There are many variants of the general SLDS and switching VAR models that are pervasive in the literature. One important example is when the dynamic matrix is shared between modes; here, the dynamics are instead distinguished based on a switching mean, such as the Markov switching stochastic volatility (MSSV) model. In the maneuvering target tracking community, it is often further assumed that the dynamic matrix is shared and *known* (due to the understood physics of the target). We explore both of these variants in the following sections.

4.3.1 Shared Dynamic Matrix, Switching Driving Noise

In many applications, the dynamics of the switching process can be described by a shared linear dynamical system matrix A; the dynamics within a given mode are then determined by some external force acting upon this LDS, and it is how this force is exerted that is mode-specific. The general form for such an SLDS is given by:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ \boldsymbol{x}_t &= A\boldsymbol{x}_{t-1} + \boldsymbol{e}_t(z_t) \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \boldsymbol{w}_t, \end{aligned} \tag{4.29}$$

where

$$\boldsymbol{e}_t(k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}) \quad \boldsymbol{w}_t \sim \mathcal{N}(0, R). \tag{4.30}$$

In this scenario, the data are generated from one dynamic matrix, A, and multiple process noise covariance matrices, $\Sigma^{(k)}$. Thus, one cannot place a MNIW prior jointly on these parameters (conditioned on $\boldsymbol{\mu}^{(k)}$) due to the coupling of the parameters in this prior. We instead consider independent priors on A, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$. We will refer to the choice of a normal prior on A, inverse-Wishart prior on $\Sigma^{(k)}$, and normal prior on $\boldsymbol{\mu}^{(k)}$ as the *N-IW-N* prior. See Appendix F.2 for details on deriving the resulting posterior distributions given these independent priors. The appendix derives the distributions assuming both mode-specific process noise parameters $\{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$ *and* a mode-specific dynamic matrix $A^{(k)}$. However, the derivations for a shared dynamic matrix A follow directly by considering data from all time steps, not just those with $z_t = k$.

Stochastic Volatility

An example of an SLDS in a similar form to that of Eq. (4.29) is the Markov switching stochastic volatility (MSSV) model. Hamilton [63] provides the seminal work in proposing Markov-switching autoregressive models for econometric modeling; applications include modeling GNP [63] and interest rates [47]. Kim [94] extends such models to the SLDS framework. The standard stochastic volatility model [152] is extended to account for regime switching by So et al. [154], resulting in the MSSV. The MSSV assumes that the log-volatilities follow an AR(1) process with a Markov switching mean.

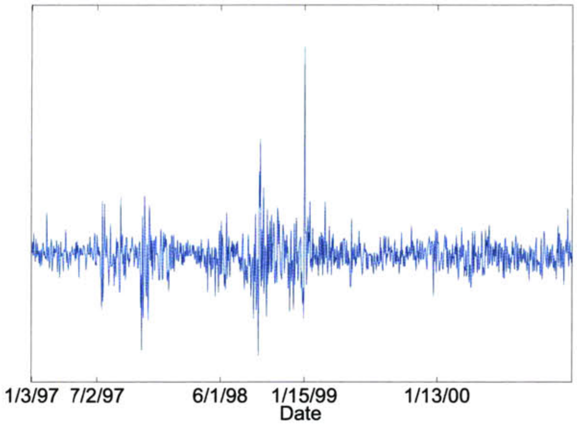


Figure 4.13. IBOVESPA stock index daily returns from 01/03/1997 to 01/16/2001.

This underlying process is observed via conditionally independent and normally distributed daily returns. Specifically, let y_t represent, for example, the daily returns of a stock index. The state x_t is then given the interpretation of log-volatilities and the resulting state space is given by [27]:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ x_t &= a x_{t-1} + e_t(z_t) \\ y_t &= u_t(x_t), \end{aligned} \tag{4.31}$$

where

$$e_t(k) \sim \mathcal{N}(\mu^{(k)}, \sigma^2) \quad u_t(x_t) \sim \mathcal{N}(0, \exp(x_t)). \tag{4.32}$$

Here, only the mean of the process noise is mode-specific. Note, however, that the measurement equation is non-linear in the state x_t. Carvalho and Lopes [27] employ a particle filtering approach to cope with these non-linearities. In [154], the MSSV is instead modeled in the log-squared-daily-returns domain such that

$$\log(y_t^2) = x_t + w_t \tag{4.33}$$

where w_t is additive, non-Gaussian noise. This noise is sometimes approximated by a moment-matched Gaussian [64], while So et al. [154] use a mixture of Gaussians approximation. The MSSV is then typically bestowed a fixed set of two or three regimes of volatility.

We test two variants of the HDP-SLDS on the IBOVESPA stock index (Sao Paulo Stock Exchange) over the period of 01/03/1997 to 01/16/2001, during which ten key world events are cited in [27] as affecting the emerging Brazilian market during this time period. The daily returns are displayed in Fig. 4.13 and the key world event are

Date	Event
07/02/1997	Thailand devalues the Baht by as much as 20%
08/11/1997	IMF and Thailand set a rescue agreement
10/23/1997	Hong Kongs stock index falls 10.4% South Korea won starts to weaken
12/02/1997	IMF and South Korea set a bailout agreement
06/01/1998	Russias stock market crashes
06/20/1998	IMF gives final approval to a loan package to Russia
08/19/1998	Russia officially falls into default
10/09/1998	IMF and World Bank joint meeting to discuss global economic crisis The Fed cuts interest rates
01/15/1999	The Brazilian government allows its currency, the Real, to float freely by lifting exchange controls
02/02/1999	Arminio Fraga is named President of Brazils Central Bank

Table 4.2. Table of 10 key world events affecting the IBOVESPA stock index (Sao Paulo Stock Exchange) over the period of 01/03/1997 to 01/16/2001, as cited by Carvalho and Lopes [27].

summarized in Table 4.2 and shown in the plots of Fig. 4.14. Use of this dataset was motivated by the work of Carvalho and Lopes [27], in which a two-mode MSSV model is assumed.

The first variant of the HDP-SLDS we consider, which we refer to as *Model A*, is simply the HDP-SLDS of Eq. (4.1) operating on the raw daily returns. The second variant, referred to as *Model B*, has a clearer interpretation as an MSSV model. Specifically, we consider a model similar to that of Eq. (4.29) and operate on log-squared daily returns. The difference between Model B and the form of the model in Eq. (4.29) is that we use a DP mixture of Gaussian measurement noise model instead of the single Gaussian model since this representation better matches the standard MSSV model. We truncate the measurement noise DP mixture to 10 components. Both models are assumed to have a one-dimensional underlying state vector (i.e., $x_t \in \mathbb{R}$.) See Table 4.3 for a summary of Model A and Model B.

For the IBOVESPA experiments, we used the same hyperparameter settings for the prior distributions on the concentration parameters α, γ, and κ as in Sec. 4.2.1-4.2.3. The prior distributions on the dynamic parameters were once again set from statistics of the data.

For Model A, where we are considering raw observations and a MNIW prior, we first pre-processed the data in the same manner as the honey bee data by centering the observations around 0 and scaling the data to be roughly within a $[-10, 10]$ dynamic range. We then set the MNIW prior with $M = 0$, $K = 1$, and $n_0 = 3$ degrees of freedom. The expected process noise covariance was set to 0.75 times the empirical covariance of the data. The IW prior on the measurement noise covariance, R, was given $r_0 = 100$

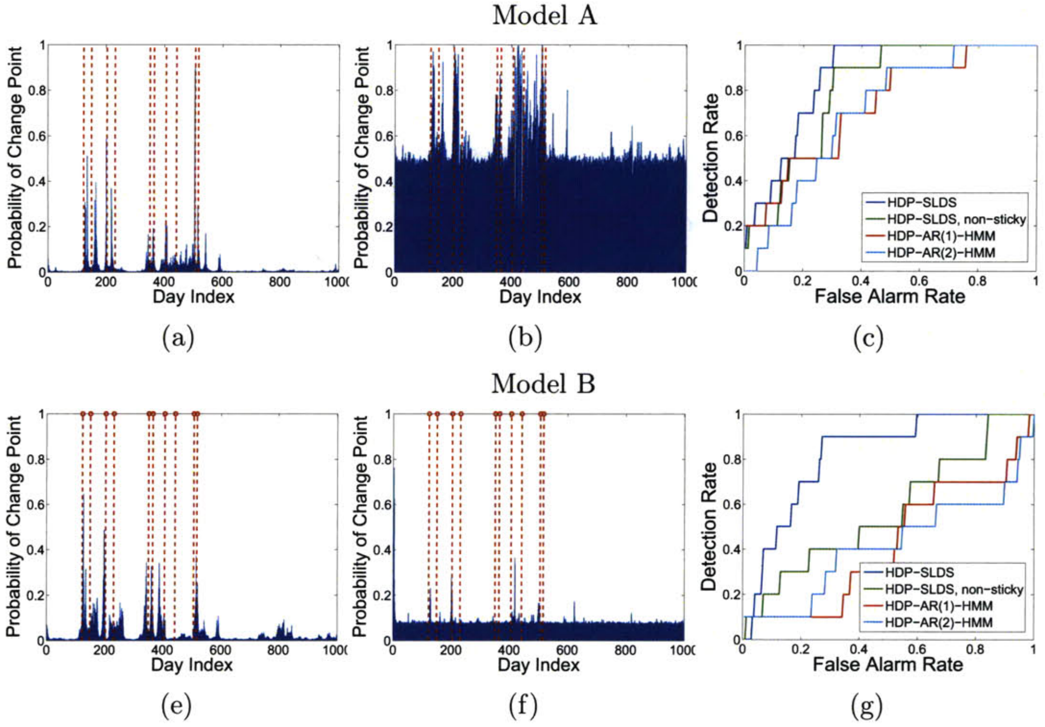
	Model A	Model B
Mode dynamics	$z_t \sim \pi_{z_{t-1}}$	$z_t \sim \pi_{z_{t-1}}$
Observation dynamics	$x_t = A^{(z_t)}x_{t-1} + e_t(z_t)$ $y_t = Cx_t + w_t$	$x_t = Ax_{t-1} + e_t(z_t)$ $y_t = Cx_t + w_t$
Noise distributions	$e_t(k) \sim \mathcal{N}(0, \Sigma^{(k)})$ $w_t \sim \mathcal{N}(0, R)$	$e_t(k) \sim \mathcal{N}(\mu^{(k)}, \Sigma^{(k)})$ $w_t \sim \sum_{\ell=1}^{\infty} \omega_\ell \mathcal{N}(0, R_\ell)$, $\boldsymbol{\omega} \sim \text{GEM}(\sigma_r)$, $R_\ell \sim \text{IW}(n_r, S_r)$
Observation type	Daily returns	Log-squared daily returns

Table 4.3. Summary of two variants of the HDP-SLDS for detecting changes in volatility of a stock index.

degrees of freedom and an expected covariance of 25. Our sampler initializes parameters from the prior, and for Model A we found it useful to set the prior around large values of R in order to avoid initial samples chattering between dynamical regimes caused by the state sequence having to account for the noise in the observations. After accounting for the residuals of the data in the posterior distribution, we typically learned $R \approx 10$. For the HDP-AR(r)-HMM's to which we compare in Fig. 4.14, we use $M = 0$, $K = 10 * I_r$, and $n_0 = 3$. Since the additional measurement noise was helpful in the Model A HDP-SLDS, we allowed for larger noise terms in the HDP-VAR(r)-HMM's as well. Namely, we set the expected covariance of the noise process to be equal to the sum of our expected process noise and measurement noise in the HDP-SLDS case (i.e., 0.75 times the empirical covariance of the data plus 25.)

For Model B, we rely on the N-IW-N prior described in Sec. 4.3.1. Since we are allowing for a mean on the process noise and dealing with log-squared daily returns, we do not perform any pre-processing of the data. For the normal prior on the dynamic parameter a, we set the mean to 0 and the covariance to 0.75 times the empirical covariance of the observations. This matches with the moments on a defined by our MNIW prior when fixing the process noise covariance to its expected value (see Eq. (2.93)). Our normal prior on $\mu^{(k)}$ is also given a 0 mean with covariance equal to 0.75 times the empirical covariance. The IW prior on the process noise $\Sigma^{(k)}$ was given 3 degrees of freedom and an expected value of 0.75 times the empirical covariance. Finally, each component of the mixture-of-Gaussian measurement noise was given an IW prior with 3 degrees of freedom and an expected value of $5 * \pi^2$, which matches with the moment-matching technique of Harvey et al. [64]. For the HDP-AR(r)-HMM's to which we compare in Fig. 4.14, we once again place a zero-mean normal prior on the dynamic parameter a with covariance set to the expected noise covariance, which in this case is equal to 0.75 times the empirical covariance plus $5 * \pi^2$ (using the same rationale as in Model A.) This IW prior on the noise parameter is given 3 degrees of freedom. As with the Model B HDP-SLDS, the mean parameter $\mu^{(k)}$ is given a normal, zero-mean prior with covariance equal to 0.75 times the empirical covariance.

The posterior probability of an HDP-SLDS inferred change point for both Model A



(a) (b) (c)

(e) (f) (g)

Figure 4.14. (a) Plot of the estimated probability of a change point on each day using 3,000 Gibbs samples for the HDP-SLDS of Eq. (4.1) on raw daily return measurements. The 10 key events are indicated with red lines. (b) Similar plot for the *non-sticky* HDP-SLDS with no bias towards self-transitions. (c) ROC curves for the HDP-SLDS, non-sticky HDP-SLDS, HDP-AR(1)-HMM, and HDP-AR(2)-HMM. (e)-(g) Analogous plots for the HDP-SLDS matched to the MSSV by using a shared dynamic matrix and allowing a mean on the mode-specific process noise and a mixture of Gaussian measurement noise model. For this matched model, we use log-squared daily returns.

and Model B is shown in Fig. 4.14(a) and Fig. 4.14(d), respectively. In Fig. 4.14(b) and Fig. 4.14(e), we display the corresponding plots for non-sticky variants of these HDP-SLDS models (i.e., with $\kappa = 0$ so that there is no bias towards mode self-transitions.) The Model A HDP-SLDS is able to infer very similar change points to those presented in [27]. Interestingly, the HDP-SLDS consistently identifies three regimes of volatility versus the assumed two-mode model of Carvalho and Lopes [27]. Without the sticky extension, the non-sticky model variant over-segments the data and rapidly switches between redundant states leading to many inferred change points that do not align with any world event. The Model B HDP-SLDS performs similarly; the non-sticky variant here once again rapidly switches between states, but not as frequently as in the Model A case. Overall, although the state of the Model B HDP-SLDS has the interpretation of log-volatilities, we see that the Model A HDP-SLDS can also capture regime-changes in the dynamics of this stock index.

In Fig. 4.14(c) and Fig. 4.14(f), the overall change-point detection performance of the Model A and Model B HDP-SLDS are compared to that of the HDP-AR(1)-HMM, HDP-AR(2)-HMM, and non-sticky HDP-SLDS. The ROC curves shown in these plots are calculated by windowing the time axis and taking the maximum probability of a change point in each window. These probabilities are then used as the confidence of a change point in that window. For both Model A and Model B, we clearly see the advantage of using an SLDS model combined with the sticky HDP-HMM prior on the mode sequence.

■ 4.3.2 Fixed Dynamic Matrix, Switching Driving Noise

There are some cases in which the dynamical model is well-defined through knowledge of the physics of the system being observed, such as simple kinematic motion. More complicated motions can typically be modeled using the same fixed dynamical model, but using a more complex description of the driving force. Returning to the model of Eq. (2.140), a generic LDS driven by an unknown control input $\boldsymbol{u}_t$ can be represented as:

$$\begin{aligned} \boldsymbol{x}_t &= A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \boldsymbol{v}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t + \boldsymbol{w}_t, \end{aligned} \tag{4.34}$$

where $\boldsymbol{v}_t \sim \mathcal{N}(0, Q)$ and $\boldsymbol{w}_t \sim \mathcal{N}(0, R)$. It is often appropriate to assume $D = 0$, as we do herein.

Maneuvering Target Tracking

The methods for modeling a maneuvering target can be primarily classified into three categories: (1) methods which approximate the non-random but unobserved control input sequence $\boldsymbol{u}_{1:T}$ as a deterministic unknown, (2) methods which model $\boldsymbol{u}_{1:T}$ as a random process, and (3) methods which use a set of dynamic systems to model typical target trajectories. For a thorough survey of maneuvering target tracking, see [144, 145].

Inference on systems modeled with deterministic unknown inputs is computationally complex, and thus modeling the control input as a random process is a common simplifying assumption. The most basic of these models is to take the control to be white noise. The constant velocity (CV) and constant acceleration (CA) models, with random walks on velocity and acceleration, respectively, are contained within this class. An immediate extension is to model the control input as a zero-mean Markov process [153]. A more realistic model is to assume that the input is a stochastic process with both temporal correlation and a time-varying, non-zero mean, such as a switching Markov process. Classical approaches, such as [119], rely on a fixing a finite set of mean values, typically a discretization of the acceleration space, and setting the probability of jumping between these modes. One can describe such a model of noisy, jump-mean control inputs by:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ \boldsymbol{x}_t &= A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t(z_t) + \boldsymbol{v}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \boldsymbol{w}_t, \end{aligned} \tag{4.35}$$

where

$$\boldsymbol{u}_t(k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}) \quad \boldsymbol{v}_t \sim \mathcal{N}(0, Q) \quad \boldsymbol{w}_t \sim \mathcal{N}(0, R). \tag{4.36}$$

Equivalently, the state dynamics can be described as

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{e}_t(z_t) \tag{4.37}$$

$$\boldsymbol{e}_t(k) \sim \mathcal{N}(B\boldsymbol{\mu}^{(k)}, B\Sigma^{(k)}B^T + Q). \tag{4.38}$$

This model can be captured by our HDP-SLDS formulation of Eq. (4.29) with a fixed dynamic matrix and mode-specific, non-zero mean process noise. Such a formulation can be viewed as an extension of the work by Caron et al. [24] in which the exogenous input is modeled as an independent noise process (i.e., no Markov structure on z_t) generated from a DP mixture model.

An alternative formulation for capturing maneuvering target dynamics is that of multiple models, which describes the targets maneuvers as switches between a set of dynamic models, e.g. CV and CA. These dynamic models can capture more coherent maneuver behavior than random processes and are well-suited to applications where target dynamics have well-defined system models with known parametrization, such as tracking civilian air traffic, tactical ballistic missiles, etc. However, for tracking hostile or noncooperative targets, such as evasive manned aircraft, the strong maneuverability and unpredictable behavior are challenging for multiple model methods which rely heavily on prior knowledge for defining the models and mode-switching probabilities. In this section, we examine the ability of the HDP-SLDS to track a highly maneuverable target. Here, we assume that the dynamic matrix A is well-modeled by either a CV or CA model, but the control input process is challenging to define. In scenarios for which a good representative set of kinematic models have not been developed, such as in tracking targets like ships and ground targets [145], the more general HDP-SLDS that additionally learns a set of mode-specific dynamic matrices may be better suited.

An Alternative Sampling Scheme

In some applications, the control input might be much lower-dimensional than the state. Harnessing the knowledge of the dynamics of the system and sampling the control input instead of the state sequence can lead to dramatic improvements in the mixing rate of the Gibbs sampler. One can sequentially block-sample $(z_t, \boldsymbol{u}_t)$, marginalizing over the state sequence $\boldsymbol{x}_{1:T}$, the transition distributions $\boldsymbol{\pi}$, and the dynamic parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$. As with the direct assignment sticky HDP-HMM sampler of Algorithm 9, this sampler relies on instantiating the global transition distribution β. We jointly sample $(z_t, \boldsymbol{u}_t)$ because a change in assignment of the mode z_t may induce a significant change in the distribution over the input $\boldsymbol{u}_t$; sampling these variables independently could result in different local modes between which it is very challenging to move. For this sampler, we assume that the measurement noise covariance R is known. Alternatively, one could interleave a step of sampling the state sequence $\boldsymbol{x}_{1:T}$ given $z_{1:T}$ and $\boldsymbol{u}_{1:T}$, and then sample R conditioned on this state sequence.

The derivation of the conditional distribution of $(z_t, \boldsymbol{u}_t)$ is outlined below, with details in Appendix E. We specifically examine a sequential node ordering for the Gibbs sampler to allow for simple updates, as will become clear in the following derivations. Let $\boldsymbol{\theta}$ denote the fixed parameters $\{A, Q, R\}$. We begin by writing

$$\begin{aligned} &p(\boldsymbol{u}_t, z_t \mid z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}, \beta, \alpha, \kappa, \lambda) \\ &\qquad = p(z_t \mid z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}, \beta, \alpha, \kappa, \lambda) p(\boldsymbol{u}_t \mid z_{1:T}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}, \lambda). \end{aligned} \tag{4.39}$$

As derived in Appendix E, we can write the conditional density of $\boldsymbol{u}_t$ for each candidate z_t as,

$$\begin{aligned} p(\boldsymbol{u}_t \mid z_t = k, z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}, \lambda) &\propto p(\boldsymbol{u}_t \mid \{\boldsymbol{u}_\tau | z_\tau = k, \tau \neq t\}, \lambda) p(\boldsymbol{y}_{1:T} \mid \boldsymbol{u}_{1:T}, \boldsymbol{\theta}) \\ &\propto \mathcal{N}^{-1}(\boldsymbol{u}_t; \hat{\Sigma}_k^{-1} \hat{\boldsymbol{\mu}}_k + \vartheta_t, \hat{\Sigma}_k^{-1} + \Lambda_t). \end{aligned} \tag{4.40}$$

Given a normal inverse-Wishart (NIW) prior on $\{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$, the posterior distribution of $\boldsymbol{u}_t$ given all other control inputs $\boldsymbol{u}_{\backslash t}$ is a Student-t distribution (see Eq. (2.87)), which we approximate by a moment-matched Gaussian $\mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k)$ (see Eq. (2.89)). The information parameters ϑ_t and Λ_t arise from marginalization of the state sequence by once again harnessing conditionally linear dynamics. Specifically, conditioning on the control input sequence simplifies the SLDS to an LDS with a deterministic control input $\boldsymbol{u}_{1:T}$. Thus, conditioning on $\boldsymbol{u}_{1:t-1,t+1:T}$ allows us to marginalize the state sequence in the following manner. We run a forward Kalman filter to pass a message from $t-2$ to $t-1$, which is updated by the local likelihood at $t-1$. A backward filter is also run to pass a message from $t+1$ to t, which is updated by the local likelihood at t. These updated messages are combined with the local dynamic $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{\theta})$ and then marginalized over $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$, resulting in the likelihood of the observation sequence $\boldsymbol{y}_{1:T}$ as a function of $\boldsymbol{u}_t$, the variable of interest. Because the sampler conditions on control inputs, the filter for this time-invariant system can be efficiently implemented by pre-computing the error covariances and then solely computing local Kalman updates at every time step. Of note is that the computational complexity is linear in the training sequence length, as well as the number of currently instantiated maneuver modes.

We similarly derive the distribution on z_t as

$$p(z_t \mid z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \theta, \beta, \alpha, \kappa, \lambda) \propto p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) C_k, \tag{4.41}$$

where $p(z_t = k \mid z_{\setminus t}, \beta, \alpha, \kappa)$ is as in Eq. (A.10) for the sticky HDP-HMM and

$$C_k = \frac{|\hat{\Sigma}_k^{-1}|^{1/2}}{|\hat{\Sigma}_k^{-1} + \Lambda_t|^{1/2}} \exp\left\{ -\frac{1}{2}\hat{\boldsymbol{\mu}}_k^T \hat{\Sigma}_k^{-1} \hat{\boldsymbol{\mu}}_k + \frac{1}{2}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \vartheta_t)^T (\hat{\Sigma}_k^{-1} + \Lambda_t)^{-1} (\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \vartheta_t) \right\}. \quad (4.42)$$

The derivation of this constant is very similar to the constant that arises in the sequential sampling of z_t described in Sec. 4.1.2.

Results

We compare the performance of our Bayesian nonparametric target tracking algorithm to that of a multiple model algorithm commonly used within the target-tracking community. Due to the exponentially growing mode-sequence hypothesis space with time, online multiple model inference methods rely on various algorithms for reducing the hypothesis space by using a *cooperation strategy*, which includes pruning, merging, and selection. Thus, these methods are theoretically suboptimal, but work well in certain situations. For each mode hypothesis, a conditional filter is run and the state estimates are then fused. Overall, the multiple model method requires model-set determination, a cooperation strategy, conditional filtering, and output processing. The standard in state-of-the-art multiple model inference algorithms, which we use in the following results, is the interacting multiple model (IMM) method [20, 115, 145].

For our Bayesian nonparametric approach, we consider the HDP-SLDS of Eq. (4.35) and take the fixed dynamic and control matrices to be:[11]

$$A = \begin{bmatrix} 1 & \Delta T & \frac{1}{2}\Delta T^2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{1}{2}\Delta T^2 \\ \Delta T \\ 0 \end{bmatrix}. \quad (4.43)$$

Here, the state consists of the x-direction position, velocity, and acceleration and we assume that we only have noisy observations of the position of the target such that $C = [1\ 0\ 0]$.

We compare this HDP-SLDS to a multiple model formulation using the standard constant velocity (CV) and constant acceleration (CA) coordinate-uncoupled maneuver models, with the state being x-direction position and velocity in the case of the CV model and x-direction position, velocity, and acceleration in the case of the CA model. The multiple model state space equations are,

$$\begin{aligned} \boldsymbol{x}_t &= \mathbf{A}^{(z_t)}\boldsymbol{x}_{t-1} + \boldsymbol{e}_t(z_t) \\ \boldsymbol{y}_t &= \mathbf{C}^{(z_t)}\boldsymbol{x}_t + \boldsymbol{w}_t. \end{aligned} \quad (4.44)$$

[11] For this scenario, we take $\boldsymbol{u}_t$ to be the control input integrated into the system over the time window $t-1$ to t. This parallels the IMM dynamics to which we compare our performance.

The system matrices for these two models are given by

$$\mathbf{A}^{(CV)} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \quad \mathbf{C}^{(CV)} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\Sigma^{(CV)} = q_{CV} \begin{bmatrix} \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix} \tag{4.45}$$

$$\mathbf{A}^{(CA)} = \begin{bmatrix} 1 & \Delta T & \frac{1}{2}\Delta T^2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{C}^{(CA)} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\Sigma^{(CA)} = q_{CA} \begin{bmatrix} \frac{1}{20}\Delta T^5 & \frac{1}{8}\Delta T^4 & \frac{1}{6}\Delta T^3 \\ \frac{1}{8}\Delta T^4 & \frac{1}{3}\Delta T^3 & \frac{1}{2}\Delta T^2 \\ \frac{1}{6}\Delta T^3 & \frac{1}{2}\Delta T^2 & \Delta T \end{bmatrix}. \tag{4.46}$$

The IMM inference algorithm requires the definition of a transition matrix P defining the probability P_{ij} of transitioning to model j given a current model i. We take

$$P = \begin{bmatrix} p_{ii} & 1-p_{ii} \\ 1-p_{ii} & p_{ii} \end{bmatrix}, \tag{4.47}$$

since we have no prior bias towards the CA model versus the CV model. Additionally, we assume that initially both models are equally likely.

It is important to understand the differences between the CV-CA multiple model formulation and our HDP-SLDS. The CV-CA multiple model formulation attempts to open up the bandwidth for maneuvers by incorporating a random walk on acceleration. However, such a formulation cannot capture the abrupt jumps in acceleration characteristic of highly maneuverable targets, whereas our formulation does. Specifically, the HDP-SLDS model we have defined takes the acceleration to be a Markov jump-mean process with a random walk noise component. Here, the parameter $\boldsymbol{\mu}^{(z_t)}$ represents the mean of this process at time t while $\Sigma^{(z_t)}$ allows for mode-specific variation of the control input realization $\boldsymbol{u}_t$. If we learn a mode with a mean of zero, our model reduces to that of the CA model (i.e. zero-mean random walk on acceleration.) When, in addition, the learned covariance $\Sigma^{(z_t)}$ and fixed process noise covariance Q are small, this model adequately describes a non-maneuvering target. However, by having the flexibility of learning modes with non-zero means, our model can account for fast changes in acceleration.

To compare the performance of the HDP-SLDS to that of the CV-CA IMM, we generated two types of simulated observations of position versus time. The first sequence (Scenario A) is a noisy version of a modulated sinusoid starting at a random phase point with measurement noise variance $R = 5 * 10^5$. The underlying position sequence has continuous derivatives so that velocity and acceleration vary smoothly. The second sequence (Scenario B) was a noisy step function generated from a three-mode Markov jump-mean model with[12] $R = 5 * 10^9$. The three modes of the model were defined by means $\{-50, 0, 50\}$ and variances $\{5, 1, 5\}$, respectively. The probability of self-transition was set to 0.99 while transitions to the other modes were equally likely. By considering both smooth and abrupt changes in acceleration, we show the flexibility of the proposed HDP-SLDS model.

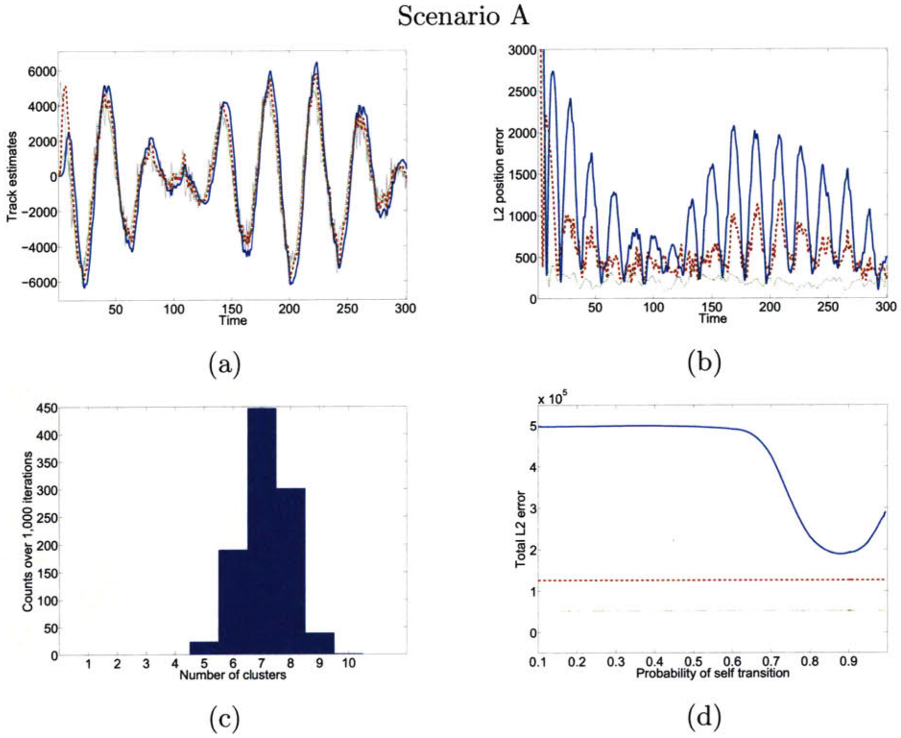
We set the HDP-SLDS and CV-CA IMM model parameters in the following manner. We use an initial error covariance $P_0 = 100 * I_3$ and step size $\Delta T = 1$. For the CV-CA IMM, we take $q_{CA} = q_{CV} = 10$, while for the HDP-SLDS, we use a small noise covariance $Q = 0.01 * I_3$ in order to encourage $\boldsymbol{u}_{1:T}$ to capture the statistical properties of the input process. For the HDP-SLDS, we place a conjugate $\mathcal{NIW}(0.001, 0, 50, 1)$ prior on the parameters $\{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$.

In the following set of results we present two methods of using the samples provided by the HDP-SLDS inference algorithm. One method involves learning the control input sequence $\boldsymbol{u}_{1:T}$ from the observation sequence $\boldsymbol{y}_{1:T}$ and then calculating Kalman smoothed state estimates given the learned input sequence. We take our estimate of $\boldsymbol{u}_{1:T}$ to simply be the average over 100 Gibbs samples. We refer to this method as the *HDP-SLDS smoother.* The batch processing of data used by the HDP-SLDS smoother is impractical in many applications. Therefore, we also present an offline-training online-tracking approach to learning a set of dynamic models that can be used within the IMM framework. Specifically, we run the HDP-SLDS sampler on training data until it is well-mixed and then examine a set of 10 samples of $(\boldsymbol{u}_{1:T}, z_{1:T})$. From each of these samples, we infer a set of parameters $\{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$ and transition densities π_k. The resulting HDP-SLDS-learned IMMs consist of CA dynamic models with different noise processes, both in terms of mean and covariance as determined by $\{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$, and by the transition probabilities π_k. The results we present for this method are state estimates averaged over the 10 parallel HDP-SLDS-learned IMMs, where the models were trained on either sinusoidal data with random phase shifts for Scenario A, or from observation sequences generated from random step function input sequences on acceleration for Scenario B.

In Fig. 4.15(d) and Fig. 4.16(d) we plot the performance of the CV-CA IMM as a function of the transition probability parameter p_{ii}. We see that the IMM exhibits strong model sensitivity to p_{ii}, while the HDP-SLDS does not depend on presetting this parameter. In the experiments for Scenarios A and B (shown in Fig. 4.15 and Fig. 4.16, respectively), we fix $p_{ii} = 0.95$ in order to consider a single "good" IMM for both of these scenarios.

In Fig. 4.15(a) and Fig. 4.16(a), we show the track estimates of position versus time for the CV-CA IMM, HDP-SLDS-learned IMMs, and HDP-SLDS smoother, as

[12] The large measurement noise setting is due to the large scale of the position observations depicted in 4.16.



(a) (b)

(c) (d)

Figure 4.15. Plots of (a) observation sequence (gray) with track estimates and (b) associated L_2 error for the CV-CA IMM (blue,—), HDP-SLDS learned IMMs (red,- -), and HDP-SLDS smoother (green,$\cdots$) on a modulated sinusoid (Scenario A). (c) Histogram of number of HDP-SLDS maneuver modes over 1,000 Gibbs iterations on the training sequence for the learned IMMs. (d) Total L_2 error versus self transition probability p_{ii} depicting the model sensitivity of the IMM as compared to the HDP-SLDS learned IMMs or HDP-SLDS smoother, which do not depend on presetting this parameter.

well as the noisy observations. The associated average L_2 position errors versus time averaged over 10 measurement realizations of the true target trajectory are plotted in Fig. 4.15(b) and Fig. 4.16(b). These plots show the performance gain of the HDP-SLDS methods over the CV-CA IMM. Relative to the CV-CA IMM performance, the HDP-SLDS-learned IMMs have a 42% average decrease in total L_2 error in the modulated sinusoid case and 52% decrease in the step function case while the HDP-SLDS smoother has decreases of 78% and 75%.

One can analyze the complexity of the inferred HDP-SLDS model by looking at the number of maneuver modes to which a significant number of observations are assigned. We histogram those modes with more than 5% of the assignments over 1,000 Gibbs iterations in Fig. 4.15(c) and Fig. 4.16(c). When the true control inputs are drawn from a small finite set, as in the step function scenario, the HDP-SLDS describes the data with fewer model components than the more complicated modulated sinusoid scenario. These results emphasize the flexibility of the HDP-SLDS approach.

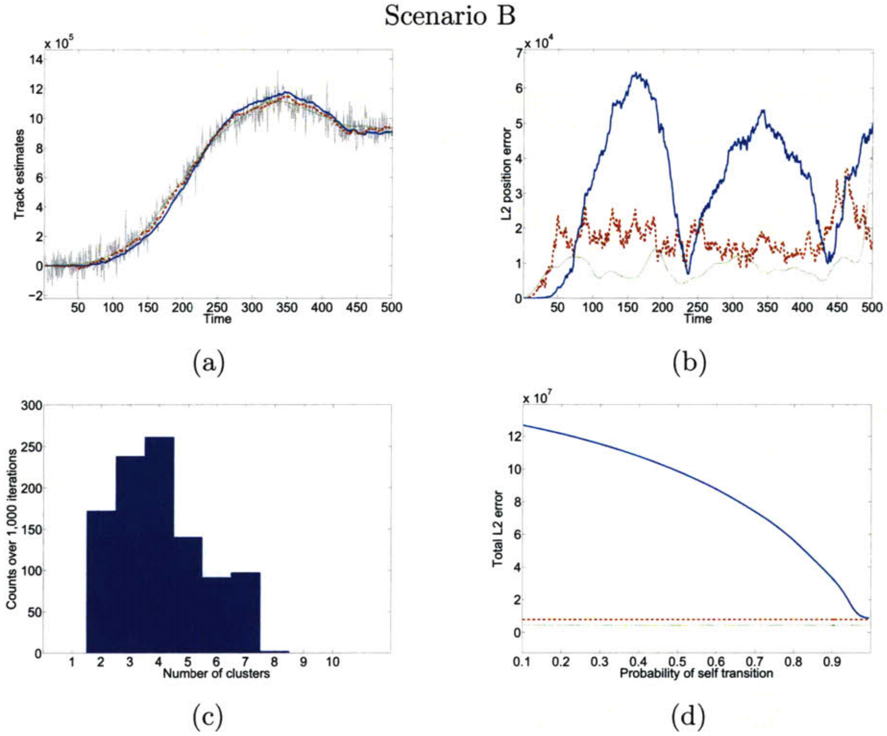


Figure 4.16. Analogous plots to those of Fig. 4.15, but for a step function control input (Scenario B).

4.4 Discussion and Future Work

In this chapter, we have addressed the problem of learning switching linear dynamical models with an unknown number of modes for describing complex dynamical phenomena. We presented a Bayesian nonparametric approach and demonstrated both the utility and versatility of the developed HDP-SLDS and HDP-AR-HMM on real applications. Using the same parameter settings, although different model choices, in one case we are able to learn changes in the volatility of the IBOVESPA stock exchange while in another case we learn segmentations of data into *waggle*, *turn-right*, and *turn-left* honey bee dances. We also described a method of applying automatic relevance determination (ARD) as a sparsity-inducing prior, leading to flexible and scalable dynamical models that allow for identification of variable order structure. The utility of this model was demonstrated on synthetic data and the honey bee dance sequences.

Following the presentation of the generic HDP-SLDS and HDP-AR-HMM models, we considered adaptations to specific forms often examined in the literature. Specifically, we developed Bayesian nonparametric variants of the Markov switching stochastic volatility model and a standard multiple model target tracking formulation. For the target tracking application, we derived a collapsed Gibbs sampler that efficiently computes smoothed state estimates from noisy observation sequences by sampling the lower-dimensional control input sequence. We showed that the parameters inferred by this sampler can be utilized in an online IMM filter and demonstrated significant gains over the fixed model set commonly used in tracking applications.

The batch processing of the Gibbs samplers derived herein may be impractical and offline-training online-tracking infeasible for certain applications. Due both to the nonlinear dynamics and uncertainty in model parameters, exact recursive estimation is infeasible. We can leverage the *conditionally linear* dynamics and use *Rao-Blackwellized particle filtering* (RBPF) [29], an efficient sequential importance sampler (SIS). Here, the particles represent samples from $p(\phi_{1:t} \mid \boldsymbol{y}_{1:t})$, where $\phi_t = \{\mathbf{A}^{(z_t)}, \Sigma^{(z_t)}\}$. Estimation of $p(\boldsymbol{x}_{1:t} \mid \phi_{1:t}, \boldsymbol{y}_{1:t})$ can then be computed using Kalman filtering. However, particle filters can suffer from a progressively impoverished particle representation, especially in the case of static parameter estimation. In our scenario, both the hyperparameters and the HDP distribution over parameters ϕ_t are static. Thus, we could modify the SIS methods, in a similar vain to *resample-move* [56] which interleaves Gibbs and particle filter steps, so that potential new values for static parameters continue to be explored over time. Another approach for an online implementation is that of decayed MCMC filtering [114]. The decayed MCMC algorithm is similar to standard MCMC methods except instead of uniformly sampling the state variables the algorithm concentrates sampling activity to the recent past, since these states are the most relevant to the current state. Decayed MCMC is guaranteed to converge to the true marginal distribution given an appropriate decay function, and has provable rates of convergence.

Another direction of future research is to develop stronger sparsity inducing priors. The ARD prior provides a simple quadratic, or L_2, penalty on the columns of A. Alternatively, one could examine the class of spike and slab priors [28, 73, 182], which place an additional *spike* of probability mass concentrated around the random variable being *exactly* zero, and have been successfully applied to both regression and factor analysis. Other examples include the Laplace prior, corresponding to the Lasso L_1 penalty [166], and the class of scale mixture of Gaussian priors presented in [23].

Overall, the formulation we developed herein represents a flexible, Bayesian nonparametric model for describing nonlinear dynamical phenomena and discovering simple underlying structures to describe time series. In the next chapter, we explore how to transfer knowledge between multiple, related time series.

Chapter 5

Sharing Features among Dynamical Systems with Beta Processes

IN many applications, one would like to discover and model dynamical behaviors which are shared among several related time series. For example, consider video or motion capture data depicting multiple people performing a number of related tasks. By jointly modeling such sequences, we may more robustly estimate representative dynamic models, and also uncover interesting relationships among activities. We specifically focus on time series where behaviors can be individually modeled via temporally independent or linear dynamical systems, and where transitions between behaviors are approximately Markovian. Examples of such *Markov jump processes* include the hidden Markov model (HMM), switching vector autoregressive (VAR) process, and switching linear dynamical system (SLDS). These models have proven useful in such diverse fields as speech recognition, econometrics, remote target tracking, and human motion capture. We have presented Bayesian nonparametric approaches to learning such models in Chapters 3 and 4, and examined some of these applications. In this chapter, our approach envisions a large *library* of behaviors, and each time series or *object* exhibits a subset of these behaviors. We then seek a framework for discovering the set of dynamic behaviors, or *features*, that each object exhibits. We particularly aim to allow flexibility in the number of total and sequence-specific behaviors, and encourage objects to share similar subsets of the large set of possible behaviors.

One can represent the set of behaviors an object exhibits via an associated list of features. A standard featural representation for N objects, with a library of K features, employs an $N \times K$ binary matrix $F = \{f_{ik}\}$. Setting $f_{ik} = 1$ implies that object i exhibits feature k. Our desiderata motivate a Bayesian nonparametric approach based on the *beta process* [67], which allows for infinitely many potential features. As shown by Thibaux and Jordan [165], integrating over the latent beta process random measure induces a predictive distribution on features equivalent to the *Indian buffet process* (IBP) of Griffiths and Ghahramani [62]. The beta process, and its connection with the IBP, are reviewed in Sec. 2.9.4. Given a sampled feature set, our model reduces to a collection of Bayesian HMMs (or SLDS) with partially shared parameters.

One approach to a Bayesian nonparametric representation of multiple time series would be to consider an extension of the sticky HDP-HMM and HDP-SLDS models of Chapters 3 and 4, respectively, in which the time series are tied together with the same set of transition and emission parameters. However, such an HDP-HMM or HDP-SLDS does not select a subset of behaviors for a given time series, but restrictively assumes that all time series share the same set of behaviors and switch among them in exactly the same manner. Another recent approach to Bayesian nonparametric modeling of Markov-switching time series is the infinite factorial HMM [171]. The infinite factorial HMM also does not solve our problem; it instead models a single time series using an infinite set of latent features, which evolve via independent Markovian dynamics. Our work focuses on modeling multiple time series and on capturing dynamical modes that are shared among the series.

Our results are obtained via an efficient and exact Markov chain Monte Carlo (MCMC) inference algorithm. In particular, we exploit the finite dynamical system induced by a fixed set of features to efficiently compute acceptance probabilities, and reversible jump birth and death proposals to explore new features. We validate our sampling algorithm using several synthetic datasets, and also demonstrate promising unsupervised segmentation of data from the CMU motion capture database [169].

■ 5.1 Describing Multiple Time Series with Beta Processes

Assume we have a set of N objects, each of whose dynamics is described by a switching vector autoregressive (VAR) process, with switches occurring according to a discrete-time Markov process. For a review of switching VAR processes, refer to Sec. 2.7.3. As we have seen in Chapter 4, such autoregressive HMMs (AR-HMMs) provide a simpler, but often equally effective, alternative to SLDS. Let $\mathbf{y}_t^{(i)}$ represent the observation vector of the i^{th} object at time t, and $z_t^{(i)}$ the latent dynamical mode. Assuming an order r switching VAR process, denoted by VAR(r), we have

$$\begin{aligned} z_t^{(i)} &\sim \pi^{(i)}_{z^{(i)}_{t-1}} \\ \mathbf{y}_t^{(i)} &= \sum_{j=1}^{r} A_{j,z_t^{(i)}} \mathbf{y}_{t-j}^{(i)} + \mathbf{e}_t^{(i)}(z_t^{(i)}) \triangleq \boldsymbol{A}_{z_t^{(i)}} \tilde{\mathbf{y}}_t^{(i)} + \mathbf{e}_t^{(i)}(z_t^{(i)}), \end{aligned} \tag{5.1}$$

where $\mathbf{e}_t^{(i)}(k) \sim \mathcal{N}(0, \Sigma_k)$, $\boldsymbol{A}_k = \begin{bmatrix} A_{1,k} & \dots & A_{r,k} \end{bmatrix}$, and $\tilde{\mathbf{y}}_t^{(i)} = \begin{bmatrix} \mathbf{y}_{t-1}^{(i)T} & \dots & \mathbf{y}_{t-r}^{(i)T} \end{bmatrix}^T$. The standard HMM with Gaussian emissions arises as a special case of this model when $\boldsymbol{A}_k = \mathbf{0}$ for all k. We refer to these VAR processes, with parameters $\theta_k = \{\boldsymbol{A}_k, \Sigma_k\}$, as *behaviors*, and use a beta process prior to couple the dynamic behaviors exhibited by different objects or sequences.

Let $\boldsymbol{f}_i$ be a vector of binary indicator variables, where f_{ik} denotes whether object i exhibits behavior k. As in Sec. 2.9.4, we can define this feature vector by utilizing the beta process prior in the following specification:

$$\begin{aligned} B \mid B_0 &\sim \mathrm{BP}(1, B_0) \\ X_i \mid B &\sim \mathrm{BeP}(B), \quad i = 1, \dots, N, \end{aligned} \tag{5.2}$$

where f_{ik} are the weights associated with the Bernoulli process realizations

$$X_i = \sum_k f_{ik} \delta_{\theta_k}. \tag{5.3}$$

As discussed in Sec. 2.9.4, marginalization of the latent beta process random measure B induces a predictive distribution on the features f_{ik} equivalent to the IBP.

Given $\boldsymbol{f}_i$, we define a *feature-constrained transition distribution* $\boldsymbol{\pi}^{(i)} = \{\pi_k^{(i)}\}$, which governs the i^{th} object's transitions among its set of dynamic behaviors. In particular, for each object i we define a doubly infinite collection of gamma-distributed random variables:

$$\eta_{jk}^{(i)} \mid \gamma, \kappa \sim \mathrm{Gamma}(\gamma + \kappa\delta(j,k), 1) \tag{5.4}$$

Here, $\delta(j,k)$ indicates the Kronecker delta function. We denote this collection of *transition variables* by $\boldsymbol{\eta}^{(i)}$, and use them to define object-specific, feature-constrained transition distributions:

$$\pi_j^{(i)} = \frac{\begin{bmatrix} \eta_{j1}^{(i)} & \eta_{j2}^{(i)} & \dots \end{bmatrix} \otimes \boldsymbol{f}_i}{\sum_{k \mid f_{ik}=1} \eta_{jk}^{(i)}} \tag{5.5}$$

Here, $\otimes$ denotes the element-wise vector product. This construction defines $\pi_j^{(i)}$ over the full set of positive integers, but assigns positive mass only at indices k where $f_{ik} = 1$, constraining the object to solely transition amongst the dynamical behaviors indicated by its feature vector.

The preceding generative process can equivalently be represented via a sample $\tilde{\pi}_j^{(i)}$ from a finite Dirichlet distribution of dimension $K_i = \sum_k f_{ik}$, containing the non-zero entries of $\pi_j^{(i)}$:

$$\tilde{\pi}_j^{(i)} \mid \boldsymbol{f}_i, \gamma, \kappa \sim \mathrm{Dir}([\gamma, \dots, \gamma, \gamma + \kappa, \gamma, \dots \gamma]) \tag{5.6}$$

The κ hyperparameter places extra expected mass on the component of $\tilde{\pi}_j^{(i)}$ corresponding to a self-transition $\pi_{jj}^{(i)}$, analogously to the sticky hyperparameter of Chapter 3. We also use the representation

$$\pi_j^{(i)} \mid \boldsymbol{f}_i, \gamma, \kappa \sim \mathrm{Dir}([\gamma, \dots, \gamma, \gamma + \kappa, \gamma, \dots] \otimes \boldsymbol{f}_i), \tag{5.7}$$

implying $\pi_j^{(i)} = \begin{bmatrix} \pi_{j1}^{(i)} & \pi_{j2}^{(i)} & \dots \end{bmatrix}$, with only a finite number of non-zero entries $\pi_{jk}^{(i)}$. This representation is really an abuse of notation since the Dirichlet distribution is

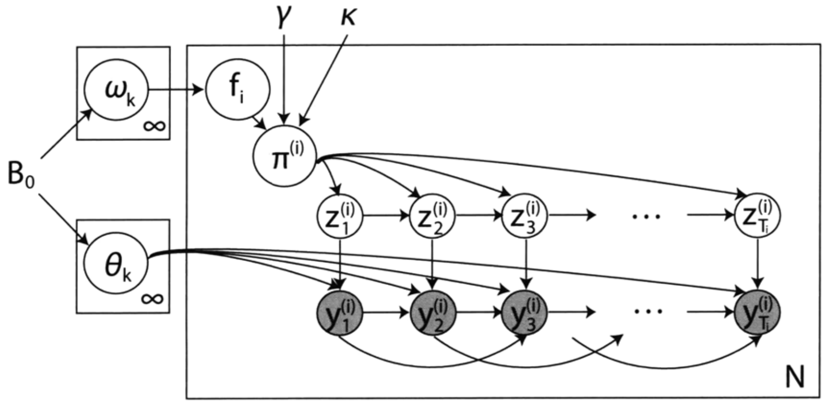


Figure 5.1. Graphical model of the IBP-AR-HMM. The beta process distributed measure $B \mid B_0 \sim \text{BP}(1, B_0)$ is represented by its masses ω_k and locations θ_k, as in Eq. (2.238). The features are then conditionally independent draws $f_{ik} \mid \omega_k \sim \text{Bernoulli}(\omega_k)$, and are used to define feature-constrained transition distributions $\pi_j^{(i)} \mid \boldsymbol{f}_i, \gamma, \kappa \sim \text{Dir}([\gamma, \dots, \gamma, \gamma + \kappa, \gamma, \dots] \otimes \boldsymbol{f}_i)$. The switching VAR dynamics are as in Eq. (5.1).

not defined for infinitely many parameters. In reality, we are simply examining a K_i-dimensional Dirichlet distribution as in Eq. (5.6). However, the notation of Eq. (5.7) is useful in reminding the reader that the indices of $\tilde{\pi}_j^{(i)}$ defined by Eq. (5.6) are not over 1 to K_i, but rather over the K_i values of k such that $f_{ik} = 1$. Additionally, this notation is useful for concise representations of the posterior distribution.

We refer to the model described in this section as the *IBP autoregressive HMM* (IBP-AR-HMM), with a graphical model representation presented in Fig. 5.1.

■ 5.2 MCMC Methods for Posterior Inference

In this section, we develop an MCMC method which alternates between resampling binary feature assignments given observations and dynamic parameters, and resampling dynamic parameters given observations and features. The sampler interleaves Metropolis-Hastings and Gibbs sampling updates, which are sometimes simplified by appropriate auxiliary variables. We leverage the fact that fixed feature assignments instantiate a set of *finite* AR-HMMs, for which dynamic programming can be used to efficiently compute marginal likelihoods. To resample the potentially infinite set of object-specific features, we introduce a new approach employing incremental "birth" and "death" proposals, improving on previous exact samplers for IBP models in the non-conjugate case [117].

5.2.1 Sampling binary feature assignments

Let $\boldsymbol{F}^{-ik}$ denote the set of all binary feature indicators excluding f_{ik}, and K_+^{-i} be the number of behaviors used by all of the other objects[1]. For notational simplicity, we assume that these behaviors are indexed by $\{1, \dots, K_+^{-i}\}$. The IBP prior differentiates between features, or behaviors, that other objects have already selected and those unique to the current object. Thus, we examine each of these cases separately. See Ex. 5.2.1 for an example illustration of the steps below.

Shared features

Given the i^{th} object's observation sequence $\mathbf{y}_{1:T_i}^{(i)}$, transition variables $\boldsymbol{\eta}^{(i)} = \eta^{(i)}_{1:K_+^{-i},1:K_+^{-i}}$, and shared dynamic parameters $\theta_{1:K_+^{-i}}$, the feature indicators f_{ik} for currently used features $k \in \{1, \dots, K_+^{-i}\}$ have the following posterior distribution:

$$p(f_{ik} \mid \boldsymbol{F}^{-ik}, \mathbf{y}_{1:T_i}^{(i)}, \boldsymbol{\eta}^{(i)}, \theta_{1:K_+^{-i}}, \alpha) \propto p(f_{ik} \mid \boldsymbol{F}^{-ik}, \alpha) p(\mathbf{y}_{1:T_i}^{(i)} \mid \boldsymbol{f}_i, \boldsymbol{\eta}^{(i)}, \theta_{1:K_+^{-i}}) \qquad (5.8)$$

Here, the IBP prior described in Sec. 2.9.4 implies that $p(f_{ik} = 1 \mid \boldsymbol{F}^{-ik}, \alpha) = m_k^{-i}/N$, where m_k^{-i} denotes the number of objects *other* than object i that exhibit behavior k. In evaluating this expression, we have exploited the exchangeability of the IBP [62], which follows directly from the beta process construction [165].

For binary random variables, Metropolis-Hastings proposals can mix faster [45] and have greater statistical efficiency [108] than standard Gibbs samplers. To update f_{ik} given $\boldsymbol{F}^{-ik}$, we thus use the posterior of Eq. (5.8) to evaluate a Metropolis-Hastings proposal which flips f_{ik} to the complement $\bar{f}$ of its current value f:

$$f_{ik} \sim \rho(\bar{f} \mid f)\delta(f_{ik}, \bar{f}) + (1 - \rho(\bar{f} \mid f))\delta(f_{ik}, f)$$
$$\rho(\bar{f} \mid f) = \min\left\{ \frac{p(f_{ik} = \bar{f} \mid \boldsymbol{F}^{-ik}, \mathbf{y}_{1:T_i}^{(i)}, \boldsymbol{\eta}^{(i)}, \theta_{1:K_+^{-i}}, \alpha)}{p(f_{ik} = f \mid \boldsymbol{F}^{-ik}, \mathbf{y}_{1:T_i}^{(i)}, \boldsymbol{\eta}^{(i)}, \theta_{1:K_+^{-i}}, \alpha)}, 1 \right\}. \qquad (5.9)$$

To compute observation likelihoods, we combine $\boldsymbol{f}_i$ and $\boldsymbol{\eta}^{(i)}$ to construct feature-constrained transition distributions $\pi_j^{(i)}$ as in Eq. (5.5), and apply a variant of the sum-product message passing algorithm of Sec. 2.6.1 for AR-HMMs that accounts for the direct correlations in the observations determined by the autoregressive process.

Unique features

An alternative approach is needed to resample the Poisson(α/N) "unique" features associated only with object i. Let $K_+ = K_+^{-i} + n_i$, where n_i is the number of unique features chosen, and define $\boldsymbol{f}_{-i} = f_{i,1:K_+^{-i}}$ and $\boldsymbol{f}_{+i} = f_{i,K_+^{-i}+1:K_+}$. The posterior distribution over n_i is then given by

[1] Some of the K_+^{-i} features may also be used by object i, but only those not unique to that object. See Example 5.2.1.

$$p(n_i \mid \boldsymbol{f}_i, \mathbf{y}_{1:T_i}^{(i)}, \boldsymbol{\eta}^{(i)}, \theta_{1:K_+^{-i}}, \alpha) \propto \frac{(\frac{\alpha}{N})^{n_i} e^{-\frac{\alpha}{N}}}{n_i!}$$

$$\iint p(\mathbf{y}_{1:T_i}^{(i)} \mid \boldsymbol{f}_{-i}, \boldsymbol{f}_{+i} = \mathbf{1}, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}_+, \theta_{1:K_+^{-i}}, \boldsymbol{\theta}_+)\, dB_0(\boldsymbol{\theta}_+) dH(\boldsymbol{\eta}_+), \quad (5.10)$$

where H is the gamma prior on transition variables $\eta_{jk}^{(i)}$, and we recall that B_0 is the base measure of the beta process. The set $\boldsymbol{\theta}_+ = \theta_{K_+^{-i}+1:K_+}$ consists of the parameters of unique features, and $\boldsymbol{\eta}_+$ the transition parameters $\eta_{jk}^{(i)}$ to or from unique features $j, k \in \{K_+^{-i} + 1 : K_+\}$. Exact evaluation of this integral is intractable due to dependencies induced by the AR-HMMs.

One early approach to approximate Gibbs sampling in non-conjugate IBP models relies on a finite truncation of the limiting Bernoulli process interpretation of the Poisson distribution [59]. That is, drawing $n_i \sim \text{Poisson}(\alpha/N)$ distribution is equivalent to setting n_i equal to the number of successes in infinitely many Bernoulli trials, each with probability of success

$$\lim_{K \to \infty} \frac{\alpha/K}{\alpha/K + N}. \quad (5.11)$$

Görür et al. [59] truncate this process and instead considers K^* Bernoulli trials with probability $(\alpha/K^*)/(\alpha/K^* + N)$. Meeds et al. [117] instead consider independent Metropolis proposals which replace the existing unique features by $n_i \sim \text{Poisson}(\alpha/N)$ new features, with corresponding parameters $\boldsymbol{\theta}_+$ drawn from the prior. For high-dimensional models like that considered in this chapter, however, such moves have extremely low acceptance rates.

We instead develop a *birth and death* reversible jump MCMC sampler [60], which proposes to either add a single new feature, or eliminate one of the existing features in $\boldsymbol{f}_{+i}$. Our proposal distribution factors as follows:

$$q(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+) = q_f(\boldsymbol{f}'_{+i} \mid \boldsymbol{f}_{+i}) q_\theta(\boldsymbol{\theta}'_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+) q_\eta(\boldsymbol{\eta}'_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{f}_{+i}, \boldsymbol{\eta}_+) \quad (5.12)$$

Let $n_i = \sum_k f_{+ik}$. The feature proposal $q_f(\cdot \mid \cdot)$ encodes the probabilities of birth and death moves: A new feature is created with probability 0.5, and each of the n_i existing features is deleted with probability $0.5/n_i$. This set of possible proposals leads to considering transitions from n_i to n'_i unique features, with $n'_i = n_i + 1$ in the case of a birth proposal, or $n'_i = n_i - 1$ in the case of a proposed feature death. Note that if the proposal from the distribution defined in Eq. (5.12) is rejected, we maintain $n'_i = n_i$ unique features. For parameters, we define our proposal using the generative model:

$$q_\theta(\boldsymbol{\theta}'_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+) = \begin{cases} b_0(\theta'_{+,n_i+1}) \prod_{k=1}^{n_i} \delta_{\theta_{+,k}}(\theta'_{+,k}), & \text{birth of feature } n_i + 1; \\ \prod_{k \neq \ell} \delta_{\theta_{+,k}}(\theta'_{+,k}), & \text{death of feature } \ell. \end{cases} \quad (5.13)$$

That is, for a birth proposal, a new parameter θ'_{+,n_i+1} is drawn from the prior and all other parameters remain the same. For a death proposal of feature j, we simply eliminate that parameter from the model. Here, b_0 is the density associated with $\alpha^{-1}B_0$. The distribution $q_\eta(\cdot \mid \cdot)$ is defined similarly, but using the gamma prior on transition variables of Eq. (5.4).

The Metropolis-Hastings acceptance probability is then given by

$$\rho(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+) = \min\{r(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+), 1\}, \tag{5.14}$$

with the acceptance ratio $r(\cdot \mid \cdot)$ derived as follows. Let us first consider a birth move in which we propose a transition from n_i to n_i+1 unique features for object i. As dictated by Eq. (5.13), the first n_i proposed components of $\boldsymbol{\theta}'_+$ and $\boldsymbol{\eta}'_+$ are equal to the previous parameters associated with those n_i features. Namely, $\theta'_{+,k} = \theta_{+,k}$ and $\eta'_{+,k} = \eta_{+,k}$ for all $k \in \{1, \dots, n_i\}$. The difference between the proposed and previous parameters arises from the fact that $\boldsymbol{\theta}'_+$ and $\boldsymbol{\eta}'_+$ contain an additional component θ'_{+,n_i+1} and η'_{+,n_i+1}, respectively, drawn from the prior distributions on these parameter spaces. Then,

$$\begin{aligned}
&r(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+) \\
&\quad = \frac{p(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{-i}, \mathbf{y}^{(i)}_{1:T_i}, \theta_{1:K^{-i}_+}, \boldsymbol{\eta}^{(i)})}{p(\boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+ \mid \boldsymbol{f}_{-i}, \mathbf{y}^{(i)}_{1:T_i}, \theta_{1:K^{-i}_+}, \boldsymbol{\eta}^{(i)})} \cdot \frac{q(\boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+)}{q(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+)} && (5.15)\\
&\quad = \frac{p(\mathbf{y}^{(i)}_{1:T_i} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}'_{+i}], \theta_{1:K^{-i}_+}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}'_+) p(\boldsymbol{f}'_{+i}) p(\boldsymbol{\theta}'_+) p(\boldsymbol{\eta}'_+)}{p(\mathbf{y}^{(i)}_{1:T_i} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}_{+i}], \theta_{1:K^{-i}_+}, \boldsymbol{\theta}_+, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}_+) p(\boldsymbol{f}_{+i}) p(\boldsymbol{\theta}_+) p(\boldsymbol{\eta}_+)} \\
&\qquad\qquad \cdot \frac{q_f(\boldsymbol{f}_{+i} \mid \boldsymbol{f}'_{+i}) q_\theta(\boldsymbol{\theta}_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+) q_\eta(\boldsymbol{\eta}_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{f}'_{+i}, \boldsymbol{\eta}'_+)}{q_f(\boldsymbol{f}'_{+i} \mid \boldsymbol{f}_{+i}) q_\theta(\boldsymbol{\theta}'_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+) q_\eta(\boldsymbol{\eta}'_+ \mid \boldsymbol{f}'_{+i}, \boldsymbol{f}_{+i}, \boldsymbol{\eta}_+)} && (5.16)
\end{aligned}$$

Noting that each component of the parameter vector $\boldsymbol{\theta}_+$ and $\boldsymbol{\eta}_+$ is drawn i.i.d., and plugging in the appropriate definitions for the proposal distributions, we have

$$\begin{aligned}
&r(\boldsymbol{f}'_{+i}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}'_+ \mid \boldsymbol{f}_{+i}, \boldsymbol{\theta}_+, \boldsymbol{\eta}_+) \\
&\quad = \frac{p(\mathbf{y}^{(i)}_{1:T_i} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}'_{+i}], \theta_{1:K^{-i}_+}, \boldsymbol{\theta}'_+, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}'_+) \text{Poisson}(n_i+1; \alpha/N) \prod_{k=1}^{n_i+1} p(\theta'_{+,k}) p(\eta'_{+,k})}{p(\mathbf{y}^{(i)}_{1:T_i} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}_{+i}], \theta_{1:K^{-i}_+}, \boldsymbol{\theta}_+, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}_+) \text{Poisson}(n_i; \alpha/N) \prod_{k=1}^{n_i} p(\theta_{+,k}) p(\eta_{+,k})} \\
&\qquad\qquad \cdot \frac{q_f(n_i \leftarrow n_i+1) \prod_{k=1}^{n_i} \delta_{\theta'_{+,k}}(\theta_{+,k}) \delta_{\eta'_{+,k}}(\eta_{+,k})}{q_f(n_i+1 \leftarrow n_i) p(\theta'_{+,n_i+1}) p(\eta'_{+,n_i+1}) \prod_{k=1}^{n_i} \delta_{\theta_{+,k}}(\theta'_{+,k}) \delta_{\eta_{+,k}}(\eta'_{+,k})}.
\end{aligned} \tag{5.17}$$

We use the notation $q_f(k \leftarrow j)$ to denote the proposal probability of transitioning from j to k unique features. Using the fact that $\theta'_{+,k} = \theta_{+,k} \in \theta_{1:K_+}$ and $\eta'_{+,k} = \eta_{+,k} \in \boldsymbol{\eta}^{(i)}$

for all $k \in \{1, \dots, n_i\}$, we can simplify the acceptance ratio to:

$$\frac{p(\mathbf{y}_{1:T_i}^{(i)} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}'_{+i}], \theta_{1:K_+}, \theta'_{+,n_i+1}, \boldsymbol{\eta}^{(i)}, \eta'_{+,n_i+1})\text{Poisson}(n_i+1; \alpha/N) q_f(n_i \leftarrow n_i+1)}{p(\mathbf{y}_{1:T_i}^{(i)} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}_{+i}], \theta_{1:K_+}, \boldsymbol{\eta}^{(i)})\text{Poisson}(n_i; \alpha/N) q_f(n_i+1 \leftarrow n_i)}. \tag{5.18}$$

The derivation of the acceptance ratio for a death move follows similarly. We compactly represent the acceptance ratio for either a birth or death move as

$$\frac{p(\mathbf{y}_{1:T_i}^{(i)} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}'_{+i}], \theta_{1:K_+}, \boldsymbol{\theta}'_{+}, \boldsymbol{\eta}^{(i)}, \boldsymbol{\eta}'_{+}) \;\text{Poisson}(n'_i \mid \alpha/N)\; q_f(\boldsymbol{f}_{+i} \mid \boldsymbol{f}'_{+i})}{p(\mathbf{y}_{1:T_i}^{(i)} \mid [\boldsymbol{f}_{-i}\, \boldsymbol{f}_{+i}], \theta_{1:K_+}, \boldsymbol{\eta}^{(i)}) \;\text{Poisson}(n_i \mid \alpha/N)\; q_f(\boldsymbol{f}'_{+i} \mid \boldsymbol{f}_{+i})}, \tag{5.19}$$

where we recall that $n'_i = \sum_k f'_{+ik}$. Because our birth and death proposals do not modify the values of existing parameters, the Jacobian term normally arising in reversible jump MCMC algorithms simply equals one.

Example 5.2.1. *Assume we have a set of four objects, and that we have finished resampling the features associated with objects 1 and 2. We also have the previous sampled feature vectors for objects 3 and 4. Let us consider the case in which this feature matrix is given by:*

$$F = \begin{bmatrix} 1 & 1 & 0 & 1 & \mid & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & \mid & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & \mid & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & \mid & 0 & 0 & 0 \end{bmatrix}$$

Then, when we resample the features for object 3 (i.e., the third row of the feature matrix), we have $K_+^{-i} = 4$ and $K_+ = 7$. The separation between the set of shared and unique features for object 3 is indicated by the dashed vertical line. For the shared features $k = 1, 2, 3, 4$, we resample f_{3k} according to the Metropolis-Hastings proposal given Eq. (5.9). For simplicity, let us assume that these features remain as they were in the previous MCMC iteration. That is, the shared feature vector is $\boldsymbol{f}_{-i} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$.

After resampling the shared features, we consider the features unique to object 3. In this case, we have $\boldsymbol{f}_{+i} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ as the unique feature vector. The feature portion of our proposal distribution, $q_f(\cdot \mid \cdot)$, defines the probability of the following possible moves:

$$\boldsymbol{f}_{+i} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \to \boldsymbol{f}'_{+i} = \begin{cases} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, & \textit{birth of feature 8;} \\ \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}, & \textit{death of feature 7;} \\ \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, & \textit{death of feature 6;} \\ \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, & \textit{death of feature 5.} \end{cases}$$

The birth move is proposed with probability 0.5 while each of the death moves has probability 0.5/3. *The associated parameter moves are given by:*

$$\boldsymbol{\theta}_+ = \{\theta_{+,1}, \theta_{+,2}, \theta_{+,3}\} \rightarrow \boldsymbol{\theta}'_+ = \begin{cases} \{\theta_{+,1}, \theta_{+,2}, \theta_{+,3}, \theta'_{+,4}\}, & \textit{birth of feature 8;} \\ \{\theta_{+,1}, \theta_{+,2}\}, & \textit{death of feature 7;} \\ \{\theta_{+,1}, \theta_{+,3}\}, & \textit{death of feature 6;} \\ \{\theta_{+,2}, \theta_{+,3}\}, & \textit{death of feature 5,} \end{cases}$$

with $\theta'_{+,4}$ *a draw from the MNIW prior. We also recall that* $\boldsymbol{\theta}_+ = \{\theta_{+,1}, \theta_{+,2}, \theta_{+,3}\} = \{\theta_5, \theta_6, \theta_7\}$, *and in a birth move we set* $\theta_8 = \theta'_{+,4}$. *The transition parameter moves are similarly defined.*

5.2.2 Sampling dynamic parameters and transition variables

Posterior updates to transition variables $\boldsymbol{\eta}^{(i)}$ and shared dynamic parameters θ_k are greatly simplified if we instantiate the mode sequences $z_{1:T_i}^{(i)}$ for each object i. We treat these mode sequences as *auxiliary variables.* Namely, these variables are sampled given the current MCMC configuration. We then resample the model parameters conditioned on the sampled auxiliary variables, after which the auxiliary variables are discarded for subsequent updates of feature assignments $\boldsymbol{f}_i$.

Mode sequences $z_{1:T_i}^{(i)}$

Given feature-constrained transition distributions $\boldsymbol{\pi}^{(i)}$ and dynamic parameters $\{\theta_k\}$, along with the observation sequence $\mathbf{y}_{1:T_i}^{(i)}$, we block sample the mode sequence $z_{1:T_i}^{(i)}$ by computing backward messages $m_{t+1,t}(z_t^{(i)}) \propto p(\mathbf{y}_{t+1:T_i}^{(i)} \mid z_t^{(i)}, \tilde{\mathbf{y}}_t^{(i)}, \boldsymbol{\pi}^{(i)}, \{\theta_k\})$, and then recursively sampling each $z_t^{(i)}$:

$$z_t^{(i)} \mid z_{t-1}^{(i)}, \mathbf{y}_{1:T_i}^{(i)}, \boldsymbol{\pi}^{(i)}, \{\theta_k\} \sim \pi_{z_{t-1}^{(i)}}^{(i)}(z_t^{(i)}) \mathcal{N}(\mathbf{y}_t^{(i)}; \boldsymbol{A}_{z_t^{(i)}} \tilde{\mathbf{y}}_t^{(i)}, \Sigma_{z_t^{(i)}}) m_{t+1,t}(z_t^{(i)}). \tag{5.20}$$

This backward message-passing, forward-sampling scheme is identical to that derived for the HDP-AR-HMM in Sec. 4.1.2, but utilizing the parameters and observations specific to object i.

Transition distributions $\pi_j^{(i)}$

We use the fact that Dirichlet priors are conjugate to multinomial observations $z_{1:T}^{(i)}$ (see Sec. 2.4.2) to derive that the posterior of $\pi_j^{(i)}$ is

$$\pi_j^{(i)} \mid \boldsymbol{f}_i, z_{1:T}^{(i)}, \gamma, \kappa \sim \mathrm{Dir}([\gamma + n_{j1}^{(i)}, \ldots, \gamma + n_{jj-1}^{(i)}, \gamma + \kappa + n_{jj}^{(i)}, \gamma + n_{jj+1}^{(i)}, \ldots] \otimes \boldsymbol{f}_i). \tag{5.21}$$

Here, $n_{jk}^{(i)}$ are the number of transitions from mode j to k in $z_{1:T}^{(i)}$. Since the mode sequence $z_{1:T}^{(i)}$ was generated from feature-constrained transition distributions, $n_{jk}^{(i)}$ will be zero for any k such that $f_{ik} = 0$. Using the definition of $\pi_j^{(i)}$ in Eq. (5.5), one can equivalently define a sample from the posterior of Eq. (5.21) by solely updating $\eta_{jk}^{(i)}$ for instantiated features:

$$\eta_{jk}^{(i)} \mid z_{1:T}^{(i)}, \gamma, \kappa \sim \text{Gamma}(\gamma + \kappa\delta(j,k) + n_{jk}^{(i)}, 1), \quad k \in \{\ell \mid f_{i\ell} = 1\}. \tag{5.22}$$

Dynamic parameters $\{\boldsymbol{A}_k, \Sigma_k\}$

We now turn to posterior updates for dynamic parameters. As with the HDP-AR-HMM of Chapter 4, we place a conjugate matrix normal inverse-Wishart (MNIW) prior on $\{\boldsymbol{A}_k, \Sigma_k\}$, comprised of an inverse-Wishart prior $\text{IW}(n_0, S_0)$ on Σ_k and a matrix-normal prior $\mathcal{MN}\left(\boldsymbol{A}_k; M, \Sigma_k^{-1}, K\right)$ on $\boldsymbol{A}_k$ given Σ_k. We consider the following sufficient statistics based on the sets $\boldsymbol{Y}_k = \{\mathbf{y}_t^{(i)} \mid z_t^{(i)} = k\}$ and $\tilde{\boldsymbol{Y}}_k = \{\tilde{\mathbf{y}}_t^{(i)} \mid z_t^{(i)} = k\}$ of observations and lagged observations, respectively, associated with behavior k:

$$\begin{aligned} S_{\tilde{y}\tilde{y}}^{(k)} &= \sum_{(t,i)|z_t^{(i)}=k} \tilde{\mathbf{y}}_t^{(i)}\tilde{\mathbf{y}}_t^{(i)^T} + \boldsymbol{K} \qquad & S_{y\tilde{y}}^{(k)} &= \sum_{(t,i)|z_t^{(i)}=k} \mathbf{y}_t^{(i)}\tilde{\mathbf{y}}_t^{(i)^T} + \boldsymbol{MK} \\ S_{yy}^{(k)} &= \sum_{(t,i)|z_t^{(i)}=k} \mathbf{y}_t^{(i)}\mathbf{y}_t^{(i)^T} + \boldsymbol{MKM}^T \qquad & S_{y|\tilde{y}}^{(k)} &= S_{yy}^{(k)} - S_{y\tilde{y}}^{(k)} S_{\tilde{y}\tilde{y}}^{-(k)} S_{y\tilde{y}}^{(k)^T}. \end{aligned} \tag{5.23}$$

It is through this pooling of observations across objects that we achieve enhanced learning of shared behaviors, especially in the presence of limited data. Analogous to the derivation outlined in Sec. 4.1.1, the posterior can then be shown to equal

$$\begin{aligned} \boldsymbol{A}_k \mid \Sigma_k, \boldsymbol{Y}_k &\sim \mathcal{MN}\left(\boldsymbol{A}_k; S_{y\tilde{y}}^{(k)} S_{\tilde{y}\tilde{y}}^{-(k)}, \Sigma_k^{-1}, S_{\tilde{y}\tilde{y}}^{(k)}\right) \\ \Sigma_k \mid \boldsymbol{Y}_k &\sim \text{IW}\left(|\boldsymbol{Y}_k| + n_0, S_{y|\tilde{y}}^{(k)} + S_0\right). \end{aligned} \tag{5.24}$$

■ 5.2.3 Sampling the IBP and Dirichlet transition hyperparameters

We additionally place priors on the Dirichlet hyperparameters γ and κ, as well as the IBP parameter α.

IBP hyperparameter α

Let $\boldsymbol{F} = \{\boldsymbol{f}_i\}$. As derived in [62], $p(\boldsymbol{F} \mid \alpha)$ can be expressed as

$$p(\boldsymbol{F} \mid \alpha) \propto \alpha^{K_+} \exp\left(-\alpha \sum_{n=1}^{N} \frac{1}{n}\right), \tag{5.25}$$

where, as before, K_+ is the number of unique features activated in $\boldsymbol{F}$. As in [59], we place a conjugate Gamma(a_α, b_α) prior on α, which leads to the following posterior distribution:

$$\begin{aligned} p(\alpha \mid \boldsymbol{F}, a_\alpha, b_\alpha) &\propto p(\boldsymbol{F} \mid \alpha) p(\alpha \mid a_\alpha, b_\alpha) \\ &\propto \alpha^{K_+} \exp\left(-\alpha \sum_{n=1}^{N} \frac{1}{n}\right) \frac{\alpha^{a_\alpha - 1} \exp(-b_\alpha \alpha)}{\Gamma(\alpha)} \\ &\propto \alpha^{a_\alpha + K_+ - 1} \exp\left(-\alpha\left(b_\alpha + \sum_{n=1}^{N} \frac{1}{n}\right)\right) \\ &= \text{Gamma}\left(a_\alpha + K_+, b_\alpha + \sum_{n=1}^{N} \frac{1}{n}\right) \end{aligned} \tag{5.26}$$

Transition hyperparameters γ and κ

Transition hyperparameters are assigned similar priors $\gamma \sim \text{Gamma}(a_\gamma, b_\gamma)$ and $\kappa \sim \text{Gamma}(a_\kappa, b_\kappa)$. Because the generative process of Eq. (5.4) is non-conjugate, we rely on Metropolis-Hastings steps which iteratively resample γ given κ, and κ given γ. Each sub-step uses a gamma proposal distribution $q_\gamma(\cdot \mid \cdot)$ or $q_\kappa(\cdot \mid \cdot)$, respectively, with fixed variance σ_γ^2 or σ_κ^2, and mean equal to the current hyperparameter value. Since a Gamma(a, b) distribution has mean a/b and variance a/b^2, these proposal distribution settings are accomplished by taking

$$q_\gamma(\cdot \mid \gamma) = \text{Gamma}\left(\frac{\gamma^2}{\sigma_\gamma^2}, \frac{\gamma}{\sigma_\gamma^2}\right) \qquad q_\kappa(\cdot \mid \kappa) = \text{Gamma}\left(\frac{\kappa^2}{\sigma_\kappa^2}, \frac{\kappa}{\sigma_\kappa^2}\right). \tag{5.27}$$

To update γ given κ, the acceptance probability is $\min\{r(\gamma' \mid \gamma), 1\}$, where the acceptance ratio is given by:

$$r(\gamma' \mid \gamma) = \frac{p(\gamma' \mid \kappa, \boldsymbol{\pi}, \boldsymbol{F}) q(\gamma \mid \gamma')}{p(\gamma \mid \kappa, \boldsymbol{\pi}, \boldsymbol{F}) q(\gamma' \mid \gamma)} = \frac{p(\boldsymbol{\pi} \mid \gamma', \kappa, \boldsymbol{F}) p(\gamma') q(\gamma \mid \gamma')}{p(\boldsymbol{\pi} \mid \gamma, \kappa, \boldsymbol{F}) p(\gamma) q(\gamma' \mid \gamma)}, \tag{5.28}$$

where we omit the hyperparameters a_γ, b_γ, and σ_γ^2 for simplicity of notation. Recalling the definition of $\tilde{\pi}_j^{(i)}$ from Eq. (5.6) and that $K_i = \sum_k f_{ik}$, the likelihood term may be written as

$$p(\boldsymbol{\pi} \mid \gamma, \kappa, \boldsymbol{F}) = \prod_i \prod_{k=1}^{K_i} p(\tilde{\pi}_k^{(i)} \mid \gamma, \kappa, \boldsymbol{f}_i) \tag{5.29}$$

$$= \prod_i \prod_{k=1}^{K_i} \left\{ \frac{\Gamma(\gamma K_i + \kappa)}{\left(\prod_{j=1}^{K_i - 1} \Gamma(\gamma)\right) \Gamma(\gamma + \kappa)} \prod_{j=1}^{K_i} \tilde{\pi}_{kj}^{(i)^{\gamma + \kappa\delta(k,j) - 1}} \right\}. \tag{5.30}$$

The ratio of the prior distributions reduces to

$$\frac{p(\gamma' \mid a_\alpha, b_\alpha)}{p(\gamma \mid a_\alpha, b_\alpha)} = \frac{\gamma'^{a_\gamma - 1}\exp\{-\gamma' b_\gamma\}}{\gamma^{a_\gamma - 1}\exp\{-\gamma b_\gamma\}} = \left(\frac{\gamma'}{\gamma}\right)^{a_\gamma - 1}\exp\{-(\gamma' - \gamma)b_\gamma\}. \tag{5.31}$$

Letting $\vartheta = \gamma^2/\sigma_\gamma^2$ and $\vartheta' = \gamma'^2/\sigma_\gamma^2$, the ratio of the proposal distributions reduces to

$$\frac{q(\gamma \mid \gamma', \sigma_\gamma^2)}{q(\gamma' \mid \gamma, \sigma_\gamma^2)} = \frac{\frac{(\gamma'/\sigma_\gamma^2)^{\vartheta'}}{\Gamma(\vartheta')}\gamma^{\vartheta'-1}\exp\{-\gamma\frac{\gamma'}{\sigma_\gamma^2}\}}{\frac{(\gamma/\sigma_\gamma^2)^{\vartheta}}{\Gamma(\vartheta)}\gamma'^{\vartheta-1}\exp\{-\gamma'\frac{\gamma}{\sigma_\gamma^2}\}} = \frac{\Gamma(\vartheta)\gamma^{\vartheta'-\vartheta-1}}{\Gamma(\vartheta')\gamma'^{\vartheta-\vartheta'-1}}\sigma_\gamma^{2(\vartheta-\vartheta')}. \tag{5.32}$$

Defining $f(\gamma) \triangleq p(\boldsymbol{\pi} \mid \gamma, \kappa, \boldsymbol{F})$, our acceptance ratio can be compactly written as

$$r(\gamma' \mid \gamma) = \frac{f(\gamma')\Gamma(\vartheta)\gamma^{\vartheta'-\vartheta-a_\gamma}}{f(\gamma)\Gamma(\vartheta')\gamma'^{\vartheta-\vartheta'-a_\gamma}}\exp\{-(\gamma' - \gamma)b_\gamma\}\sigma_\gamma^{2(\vartheta-\vartheta')}. \tag{5.33}$$

The Metropolis-Hastings sub-step for resampling κ given γ follows similarly. In this case, however, the likelihood terms simplifies to

$$f(\kappa) \triangleq \prod_i \frac{\Gamma(\gamma K_i + \kappa)^{K_i}}{\Gamma(\gamma + \kappa)^{K_i}} \prod_{j=1}^{K_i} \tilde{\pi}_{jj}^{(i)\gamma+\kappa-1} \propto p(\boldsymbol{\pi} \mid \gamma, \kappa, \boldsymbol{F}). \tag{5.34}$$

The resulting MCMC sampler for the IBP-AR-HMM is summarized in Algorithm 17.[2]

5.3 Synthetic Experiments

To test the ability of the IBP-AR-HMM to discover shared dynamics, we generated five time series that switched between AR(1) models

$$y_t^{(i)} = a_{z_t^{(i)}} y_{t-1}^{(i)} + e_t^{(i)}(z_t^{(i)}) \tag{5.35}$$

with $a_k \in \{-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8\}$ and process noise covariance Σ_k drawn from an IW(3, 0.5) prior. The object-specific features, shown in Fig. 5.2(b), were sampled from a truncated IBP [62] using $\alpha = 10$ and then used to generate the observation sequences of Fig. 5.2(a) (colored by the true mode sequences). Each row of the feature matrix corresponds to one of the five time series, and the columns represent the different autoregressive models with a white square indicating that a given time series uses that dynamical regime. Here, the columns are ordered so that the first feature corresponds to an autoregressive model defined by a_1, and the ninth feature corresponds to that of a_9. The resulting feature matrix estimated over 10,000 MCMC samples is shown in Fig. 5.2(c). Each of the 10,000 estimated feature matrices is produced from an MCMC sample of the mode sequences that are first mapped to the ground truth

[2] Note that Algorithm 18 is embedded within Algorithm 17

```
Given a previous set of object-specific transition variables $\{\boldsymbol{\eta}^{(i)}\}^{(n-1)}$, the dynamic parameters $\{\boldsymbol{A}_k, \Sigma_k\}^{(n-1)}$, and features $\boldsymbol{F}^{(n-1)}$:
 1. Set $\{\boldsymbol{\eta}^{(i)}\} = \{\boldsymbol{\eta}^{(i)}\}^{(n-1)}$, $\{\boldsymbol{A}_k, \Sigma_k\} = \{\boldsymbol{A}_k, \Sigma_k\}^{(n-1)}$, and $\boldsymbol{F} = \boldsymbol{F}^{(n-1)}$.
 2. From the feature matrix $\boldsymbol{F}$, create count vector $\boldsymbol{m} = [\ m_1 \quad m_2 \quad \dots \quad m_{K_+}\ ]$,
    with $m_k$ representing the number of objects possessing feature $k$.
 3. For each $i \in \{1, \dots, N\}$, sample features as follows:
    (a) Set $\boldsymbol{m}^{-i} = \boldsymbol{m} - \boldsymbol{f}_i$, and reorder columns of $\boldsymbol{F}$ so that the $K_+^{-i}$ columns with
        $m_k^{-i} > 0$ appear first. Appropriately relabel indices of $\{\boldsymbol{A}_k, \Sigma_k\}$ and $\{\boldsymbol{\eta}^{(i)}\}$.
    (b) For each shared feature $k \in \{1, \dots, K_+^{-i}\}$, set $f = f_{ik}$ and:
        i. Consider $f_{ik} \in \{0, 1\}$ and:
           A. Create feature-constrained transition distributions:
              $$\pi_j^{(i)} \propto [\ \eta_{j1}^{(i)} \quad \eta_{j2}^{(i)} \quad \dots \quad \eta_{jK_+}^{(i)}\ ] \otimes \boldsymbol{f}_i$$
           B. Compute likelihood $\ell_{f_{ik}}\left(\mathbf{y}_{1:T_i}^{(i)}\right) \triangleq p\left(\mathbf{y}_{1:T_i}^{(i)} \mid \boldsymbol{\pi}^{(i)}, \{\boldsymbol{A}_k, \Sigma_k\}\right)$ using a
              variant of the sum-product algorithm described in Sec. 2.6.1.
        ii. Compute
            $$\rho^* = \frac{m_k^{-i}}{N - m_k^{-i}} \cdot \frac{\ell_1\left(\mathbf{y}_{1:T_i}^{(i)}\right)}{\ell_0\left(\mathbf{y}_{1:T_i}^{(i)}\right)} \quad \text{and set} \quad \rho(\bar{f} \mid f) = \begin{cases} \min\{\rho^*, 1\}, & f = 0; \\ \min\{1/\rho^*, 1\}, & f = 1. \end{cases}$$
        iii. Sample $f_{ik} \sim \rho(\bar{f} \mid f)\delta(f_{ik}, \bar{f}) + (1 - \rho(\bar{f} \mid f))\delta(f_{ik}, f)$.
    (c) Let $\boldsymbol{f}_i$' $= \boldsymbol{f}_i$ and calculate the number of unique features $n_i = K_+ - K_+^{-i}$.
        i. Propose a birth or death move, each with probability 0.5.
           • Birth: sample $\{\theta'_{+,n_i+1}, \eta_{+,n_i+1}\}$ from their priors and set $f'_{i,n_i+1} = 1$,
             $n'_i = n_i + 1$.
           • Death: sample $\ell \sim \text{uniform}[K_+^{-i} + 1 : K_+]$ and set $f'_{i\ell} = 0$, $n'_i = n_i - 1$.
        ii. Compute likelihoods $\ell_{\boldsymbol{f}_i}\left(\mathbf{y}_{1:T_i}^{(i)}\right)$ and $\ell_{\boldsymbol{f}'_i}\left(\mathbf{y}_{1:T_i}^{(i)}\right)$ of data under the
            previous and proposed models, respectively.
        iii. Keep ($\zeta = 1$) or discard ($\zeta = 0$) proposed model by sampling:
             $$\zeta \sim \text{Ber}(\rho) \quad \rho = \min\left\{\frac{\ell_{\boldsymbol{f}_i}\left(\mathbf{y}_{1:T_i}^{(i)}\right)\text{Poisson}(n'_i \mid \frac{\alpha}{N})q_f(n_i \leftarrow n'_i)}{\ell_{\boldsymbol{f}'_i}\left(\mathbf{y}_{1:T_i}^{(i)}\right)\text{Poisson}(n_i \mid \frac{\alpha}{N})q_f(n'_i \leftarrow n_i)}, 1\right\}.$$
    (d) Set $\boldsymbol{m} = \boldsymbol{m}^{-i} + \boldsymbol{f}_i$. Remove columns for which $m_k = 0$, and appropriately
        redefine the dynamic parameters $\{\boldsymbol{A}_k, \Sigma_k\}$ and transition variables $\{\boldsymbol{\eta}^{(i)}\}$.
 4. Resample dynamic parameters $\{\boldsymbol{A}_k, \Sigma_k\}$ and transition variables $\{\boldsymbol{\eta}^{(i)}\}$ using the
    auxiliary variable sampler of Algorithm 18.
 5. Fix $\{\boldsymbol{\eta}^{(i)}\}^{(n)} = \{\boldsymbol{\eta}^{(i)}\}$, $\{\boldsymbol{A}_k, \Sigma_k\}^{(n)} = \{\boldsymbol{A}_k, \Sigma_k\}$, and $\boldsymbol{F}^{(n)} = \boldsymbol{F}$.
```

Algorithm 17. IBP-AR-HMM MCMC sampler.

```
Given the feature-restricted transition distributions $\boldsymbol{\pi}^{(i)}$ and dynamic parameters
$\{\boldsymbol{A}_k, \Sigma_k\}$, update the parameters as follows:

1. For each $i \in \{1, \dots, N\}$:
   (a) Block sample $z_{1:T_i}^{(i)}$ as follows:
       i. For each $k \in \{1, \dots, K_+\}$, initialize messages to $m_{T+1,T}^{(i)}(k) = 1$.
       ii. For each $t \in \{T_i, \dots, 1\}$ and $k \in \{1, \dots, K_+\}$, compute
           $m_{t,t-1}^{(i)}(k) = \sum_{j=1}^{K} \pi_k^{(i)}(j) \mathcal{N}\left(\mathbf{y}_t^{(i)}; \boldsymbol{A}_j \tilde{\mathbf{y}}_t^{(i)}, \Sigma_j\right) m_{t+1,t}^{(i)}(j).$
       iii. Working sequentially forward in time, and starting with transitions
            counts $n_{jk}^{(i)} = 0$:
            A. Sample a mode assignment $z_t^{(i)}$ as:
               $z_t^{(i)} \sim \sum_{k=1}^{K_+} \pi_{z_{t-1}^{(i)}}^{(i)}(k) \mathcal{N}\left(\mathbf{y}_t^{(i)}; \boldsymbol{A}_k \tilde{\mathbf{y}}_t^{(i)}, \Sigma_k\right) m_{t+1,t}^{(i)}(k) \delta\left(z_t^{(i)}, k\right).$
            B. Increment $n_{z_{t-1}^{(i)} z_t^{(i)}}^{(i)}$.
       Note that $\pi_j^{(i)}(k)$ is zero for any $k$ such that $f_{ik} = 0$, implying that $z_t^{(i)} = k$
       will never be sampled (as desired). Considering all $K_+$ indices simply allows
       for efficient matrix implementation.
   (b) For each $(j, k) \in \{1, \dots, K_+\} \times \{1, \dots, K_+\}$, sample
       $\eta_{jk}^{(i)} \mid \gamma \sim \text{Gamma}(1, \gamma + \kappa\delta(j, k) + n_{jk}^{(i)}).$

2. For each $k \in \{1, \dots, K_+\}$:
   (a) Form $\boldsymbol{Y}_k = \{\mathbf{y}_t^{(i)} | z_t^{(i)} = k\}$ and $\tilde{\boldsymbol{Y}}_k = \{\tilde{\mathbf{y}}_t^{(i)} | z_t^{(i)} = k\}$ and compute $S_{\tilde{y}\tilde{y}}^{(k)}$, $S_{y\tilde{y}}^{(k)}$,
       $S_{yy}^{(k)}$, and $S_{y|\tilde{y}}^{(k)}$ as in Eq. (5.23).
   (b) Sample dynamic parameters:
       $\Sigma_k \sim \text{IW}\left(\sum_{i=1}^{N} n_{k\cdot}^{(i)} + n_0, S_{y|\tilde{y}}^{(k)} + S_0\right)$
       $\boldsymbol{A}_k \mid \Sigma_k \sim \mathcal{MN}\left(\boldsymbol{A}_k; S_{y\tilde{y}}^{(k)} S_{\tilde{y}\tilde{y}}^{-(k)}, \Sigma_k^{-1}, S_{\tilde{y}\tilde{y}}^{(k)}\right).$
```

Algorithm 18. IBP-AR-HMM auxiliary variable sampler for updating transition and dynamic parameters.

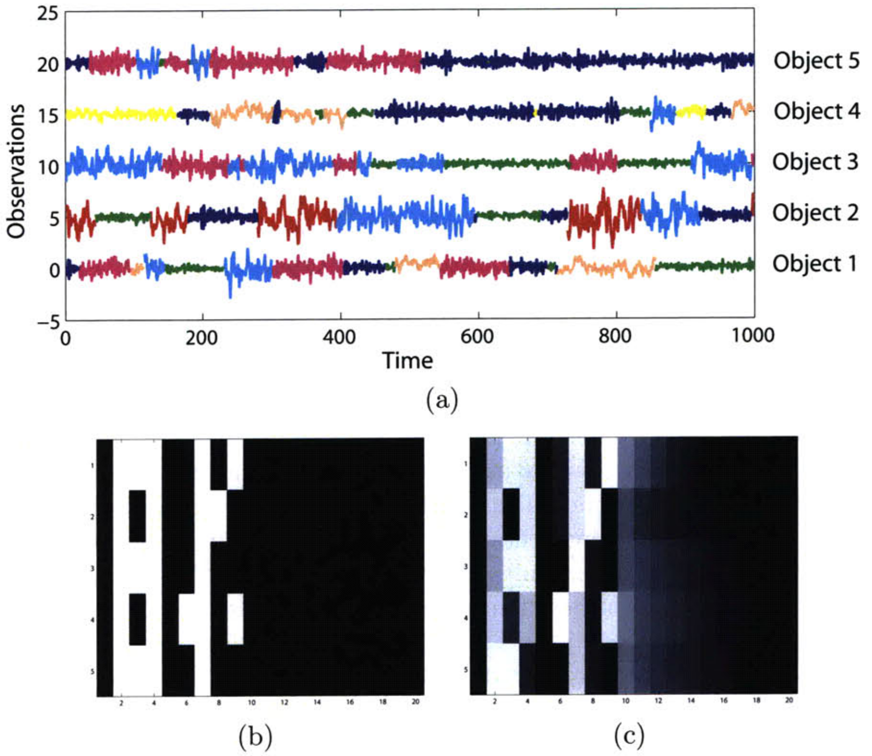


Figure 5.2. (a) Observation sequences for each of 5 switching AR(1) time series colored by true mode sequence, and offset for clarity. Images of the (b) true feature matrix of the five objects and (c) estimated feature matrix averaged over 10,000 MCMC samples taken from 100 trials every 10th sample. Each row corresponds to a different object, and each column a different autoregressive model. White indicates active features. Although the true model is defined by only 9 possible dynamical modes, we show 20 columns in order to display the "tail" of the IBP-AR-HMM estimated matrix resulting from samples that incorporated additional dynamical modes (events that have positive probability of occurring, as defined by the IBP prior.) The estimated feature matrices are produced from mode sequences mapped to the ground truth labels according to the minimum Hamming distance metric, and selecting modes with more than 2% of the object's observations.

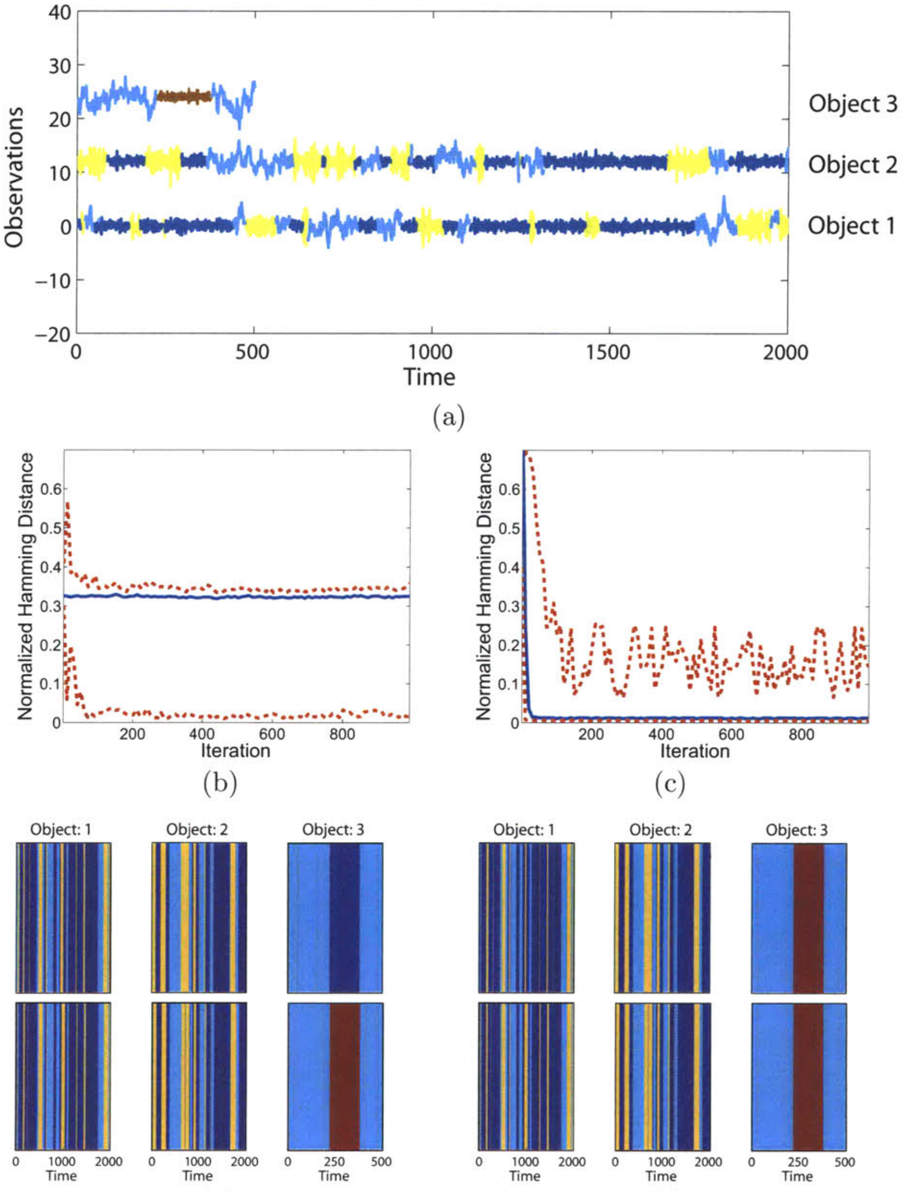
labels according to the minimum Hamming distance metric. We then only maintain inferred modes with more than 2% of the object's observations. Comparing to the true feature matrix, we see that our model is indeed able to discover most of the underlying latent structure of the time series despite the challenging setting defined by the close autoregressive coefficients. The most commonly missed feature is the use of a_4 by the fifth time series. This fifth time series is the top-most displayed in Fig. 5.2(a), and the dynamical mode defined by a_4 is shown in green. We see that this mode is used very infrequently, making it challenging to distinguish. Due to the nonparametric nature of the model, we also see a "tail" in the estimated matrix because of the (infrequent) incorporation of additional dynamical modes.

One might propose, as an alternative to the IBP-AR-HMM, the use of an architecture based on the hierarchical Dirichlet process of [162]; specifically we could use the HDP-AR-HMMs of Chapter 4 tied together with a shared set of transition and dynamic parameters. To demonstrate the difference between these models, we generated data for three switching AR(1) processes. The first two objects, with four times the data points of the third, switched between dynamical modes defined by $a_k \in \{-0.8, -0.4, 0.8\}$ and the third object used $a_k \in \{-0.3, 0.8\}$. The results shown in Fig. 5.3 indicate that the multiple HDP-AR-HMM model, which assumes all objects share *exactly* the same transition matrices and dynamic parameters, typically describes the third object using $a_k \in \{-0.4, 0.8\}$ since this assignment better matches the parameters defined by the other (lengthy) time series. This common grouping of two distinct dynamical modes leads to the large median and 90th Hamming distance quantiles shown in Fig. 5.3(b). The IBP-AR-HMM, on the other hand, is better able to distinguish these dynamical modes (see Fig. 5.3(c)) since the penalty in not sharing a behavior is only in the feature matrix; once a unique feature is chosen, it does not matter how the object chooses to use it. Example segmentations representative of the median Hamming distance error are shown in Fig. 5.3(d)-(e). These results illustrate that the IBP-based feature model emphasizes choosing behaviors rather than assuming all objects are performing minor variations of the same dynamics.

For the experiments above, we placed a Gamma(1, 1) prior on α and γ, and a Gamma(100, 1) prior on κ. The gamma proposals used $\sigma_\gamma^2 = 1$ and $\sigma_\kappa^2 = 100$ while the MNIW prior was given $M = 0$, $K = 0.1 * I_d$, $n_0 = d + 2$, and S_0 set to 0.75 times the empirical variance of the joint set of first difference observations. At initialization, each time series was segmented into five contiguous blocks, with feature labels unique to that sequence.

5.4 Motion Capture Experiments

The linear dynamical system is a common model for describing simple human motion [69], and the more complicated SLDS has been successfully applied to the problem of human motion synthesis, classification, and visual tracking [132, 133]. Other approaches develop non-linear dynamical models using Gaussian processes [179] or based on a collection of binary latent features [159]. However, there has been little effort in jointly segmenting and identifying common dynamic behaviors amongst a set of *multiple* motion capture (MoCap) recordings of people performing various tasks. The IBP-AR-HMM provides an ideal way of handling this problem. One benefit of the proposed model, versus the standard SLDS, is that it does not rely on manually specifying the set of possible behaviors. As an illustrative example, we examined a set of six CMU MoCap exercise routines [169], three from Subject 13 and three from Subject 14. Each of these routines used some combination of the following motion categories: running in place, jumping jacks, arm circles, side twists, knee raises, squats, punching, up and down, two variants of toe touches, arch over, and a reach out stretch.



(a)

(b)

(c)

(d)

(e)

Figure 5.3. (a) Observation sequences for each of 3 switching AR(1) time series colored by true mode sequence, and offset for clarity. The first and second sequences are four times as long as the third. (b)-(c) Focusing solely on the third time series, the median (solid blue) and 10^{th} and 90^{th} quantiles (dashed red) of Hamming distance between the true and estimated mode sequence over 1000 trials are displayed for the multiple HDP-AR-HMM model and the IBP-AR-HMM, respectively. (d)-(e) Examples of typical segmentations into behavior modes for the three objects at MCMC iteration 1000 for the two models. The top and bottom panels display the estimated and true sequences, respectively, and the color coding corresponds exactly to that of (a). For example, object 3 switches between two modes colored by cyan and maroon.

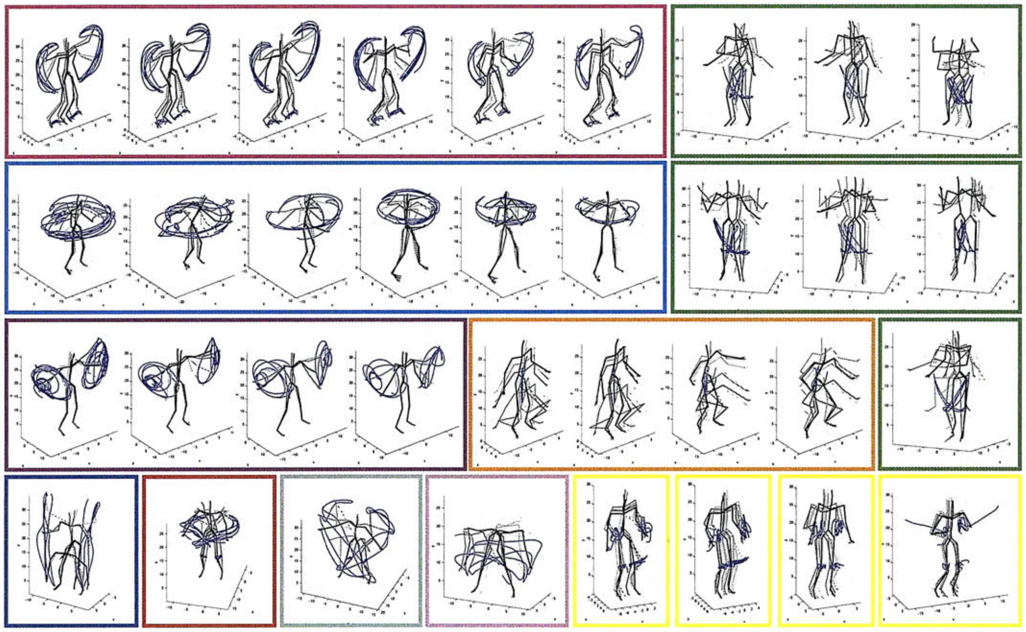


Figure 5.4. Each skeleton plot displays the trajectory of a learned contiguous segment of more than 2 seconds. To reduce the number of plots, we preprocessed the data to bridge segments separated by fewer than 300 msec. The boxes group segments categorized under the same behavior label, with the color indicating the true behavior label (allowing for analysis of split behaviors). Skeleton rendering done by modifications to Neil Lawrence's Matlab MoCap toolbox [105].

From the set of 62 position and joint angles, we selected the following set of 12 measurements deemed most informative for the gross motor behaviors we wish to capture: one body torso position, two waist angles, one neck angle, one set of right and left shoulder angles, the right and left elbow angles, one set of right and left hip angles, and one set of right and left ankle angles. As with the speaker diarization application of Sec. 3.5, we block average and downsample the data. The CMU MoCap data is recorded at a rate of at 120 frames per second, and we use a window size of 12 in our preprocessing. We additionally scale each component of the observation vector so that the empirical variance on the concatenated set of first difference measurements is 1. Using these measurements, the prior distributions were set exactly as in the synthetic data experiments except the scale matrix, S_0, of the MNIW prior which was set to $5 \cdot I_{12}$ (i.e., five times the empirical covariance of the preprocessed first difference observations, and maintaining only the diagonal.) This setting allows more variability in the observed behaviors. We ran 25 chains of the sampler for 20,000 iterations and then examined the chain whose segmentation minimized the expected Hamming distance to the set of segmentations from all chains over iterations 15,000 to 20,000. This method of choosing an MCMC sample is described in more detail in Sec. 3.5.

The resulting MCMC sample is displayed in Fig. 5.4. Each skeleton plot depicts the

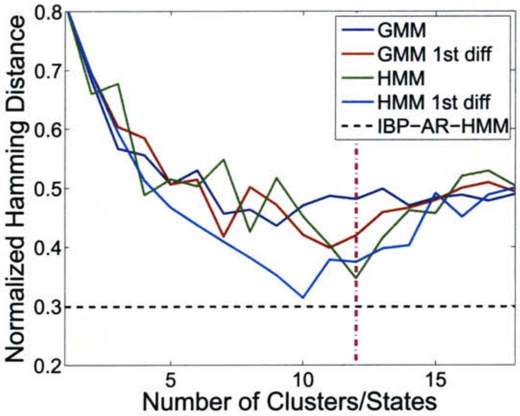


Figure 5.5. Hamming distance versus number of GMM clusters / HMM states on raw observations (blue/green) and first-difference observations (red/cyan), with the IBP-AR-HMM segmentation (black, horizontal dashed) and true feature count (magenta, vertical dashed) shown for comparison. Results are for the most-likely of 10 EM initializations using Kevin Murphy's HMM Matlab toolbox [123].

trajectory of a learned contiguous segment of more than two seconds, and boxes group segments categorized under the same behavior label by our algorithm. The color of the box indicates the true behavior label. From this plot we can infer that although some true behaviors are split into two or more categories by our algorithm, the IBP-AR-HMM shows a clear ability to find common motions. Specifically, the IBP-AR-HMM has successfully identified and grouped examples of jumping jacks (magenta), side twists (bright blue), arm circles (dark purple), squats (orange), and various motion behaviors that appeared in only one movie (bottom left four skeleton plots.) The split behaviors shown in green and yellow correspond to the true motion categories of knee raises and running, respectively, and the splits can be attributed to the two subjects performing the same motion in a distinct manner. For the knee raises, one subject performed the exercise while slightly twisting the upper in a counter-motion to the raised knee (top three examples) while the other subject had significant side-to-side upper body motion (middle three examples). For the running motion category, the splits also tended to correspond to varying upper body motion such as running with hands in or out of sync with knees. One example (bottom right) was the subject performing a lower-body run partially mixed with an upper-body jumping jack/arm flapping motion (an obviously confused test subject.) See Sec. 5.5 for further discussion of the IBP-AR-HMM splitting phenomenon.

We compare our MoCap performance to the Gaussian mixture model (GMM) method of Barbič et al. [7] using expectation maximization (EM) initialized with k-means. Barbič et al. [7] also present an approach based on probabilistic principle component analysis (PCA), but this method focuses primarily on change-point detection rather than behavior clustering. As further comparisons, we look at a GMM on first difference observations, and an HMM on both data sets. In Fig. 5.5, we analyze the ability of

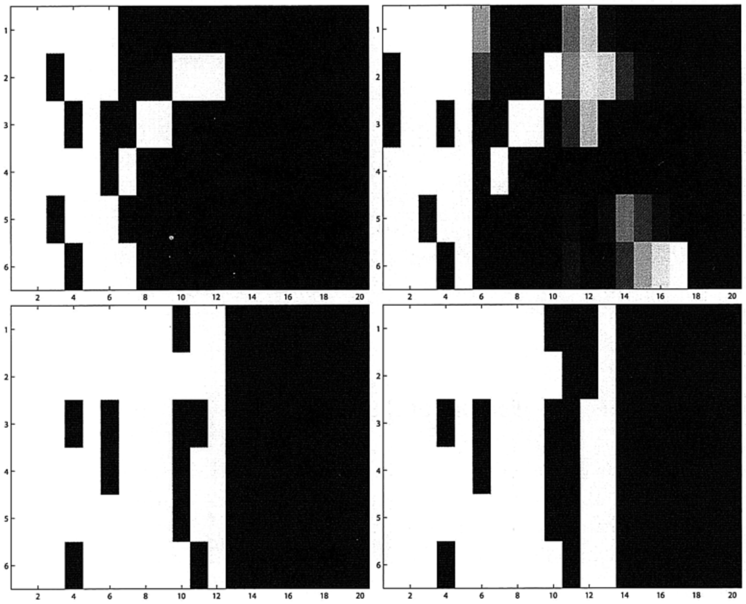


Figure 5.6. Feature matrices associated with the true MoCap sequences (top-left), IBP-AR-HMM estimated sequences over iterations 15,000 to 20,000 (top-right), and MAP assignment of the GMM (bottom-left) and HMM (bottom-right) using first-difference observations and 12 clusters/states.

the IBP-AR-HMM, as compared to the defined GMMs and HMMs, in providing accurate labelings of the individual frames of the six movie clips[3]. Specifically, we plot the Hamming distance between the true and estimated frame labels versus the number of GMM clusters and HMM states, using the most-likely of 10 initializations of EM. We also plot the Hamming distance corresponding the IBP-AR-HMM MCMC sample depicted in Fig. 5.4, demonstrating that the IBP-AR-HMM provides more accurate frame labels than any of these alternative approaches over a wide range of mixture model settings. The estimated feature matrices for the IBP-AR-HMM and the GMM and HMM on first difference observations are shown in Fig. 5.6. The figure displays the matrix associated with the MAP label estimate in the case of the GMM and HMM, and an estimate based on MCMC samples from iterations 15,000 to 20,000 for the IBP-AR-HMM. For the GMM and HMM, we consider the case when the number of Gaussian mixture components or the number of HMM states is set to the true number of behaviors, namely 12. By pooling all of the data, the GMM and HMM approaches assume that each object exhibits the same structure; the results of this assumption can be seen in the strong bands of white implying sharing of behavior between the time series. The IBP-AR-HMM estimated feature matrix, on the other hand, provides a much better match to the true matrix by allowing for sequence-specific variability. For example,

[3] The ability to accurately label the frames of a large set of movies is useful for tasks such as querying an extensive MoCap database (such as that of CMU) without relying on manual labeling of the movies.

this ability is indicated by the special structure of features in the upper right portion of the true feature matrix that is mostly captured in the IBP-AR-HMM estimated feature matrix, but is not present in those of the GMM or HMM. We do, however, note a few IBP-AR-HMM merged and split behaviors. Overall, we see that in addition to producing more accurate segmentations of the MoCap data, the IBP-AR-HMM provides a superior ability to discover the shared feature structure.

■ 5.5 Discussion and Future Work

Utilizing the beta process, we developed a coherent Bayesian nonparametric framework for discovering dynamical features common to multiple time series. This formulation allows for object-specific variability in how the dynamical behaviors are used. We additionally developed a novel exact sampling algorithm for non-conjugate IBP models. The utility of our IBP-AR-HMM was demonstrated both on synthetic data, and on a set of MoCap sequences where we showed performance exceeding that of alternative methods. Although we focused on switching VAR processes, our approach could be equally well applied to HMMs, and to a wide range of other switching dynamical systems such as the SLDS of Chapter 4.

One area of future work is to develop split-merge proposals to further improve mixing rates in high-dimensions. Although the block initialization of the time series helps with the issue of splitting merged behaviors (analogous to the issues with splitting merged speakers discussed in Chapter 3), it does not fully solve the problem and cannot be relied upon in datasets with more irregular switching patterns than the MoCap data we considered. Additionally, splitting a single true behavior into multiple estimated behaviors often occurred, and is related to the issue of splitting a single true speaker in Chapter 3. The root of the splitting issue is two-fold. One is due to the mixing rate of the sampler. The second, unlike in the case of merging behaviors, is due to modeling issues. Our model assumes that the dynamic behavior parameters (i.e., the VAR process parameters) are identical between time series and do not change over time. This assumption can be problematic in grouping related dynamic behaviors, and might be addressed via hierarchical models of behaviors or by ideas similar to those of the *dependent Dirchlet process* [61, 111] that allows for time-varying parameters.

Overall, the MoCap results appeared to be fairly robust to examples of only slightly dissimilar behaviors (e.g., squatting to different levels, twisting at different rates, etc.) However, in cases such as the running motion where only portions of the body moved in the same way while others did not, we tended to split the behavior group. This observation motivates examination of *local partition processes* [37, 38] rather than *global partition processes*. That is, our current model assumes that the grouping of observations into behavior categories occurs along all components of the observation vector rather than just a portion (e.g., lower body measurements.) Allowing for greater flexibility in the grouping of observation vectors becomes increasingly important in high dimensions.

Chapter 6

Contributions and Recommendations

WE begin with a summary of the overriding themes and principal contributions presented in the preceding chapters. Throughout the course of these chapters, we developed a flexible Bayesian framework for learning Markov switching processes to describe a wide variety of datasets; these models provide the foundation for many other interesting extensions and analyses, which we highlight following our summary of contributions.

6.1 Summary of Methods and Contributions

As we have demonstrated in this thesis, many complex dynamical phenomena, such as human motion, the dance of honey bees, trends in stock indices, and conference audio can be modeled via Markov switching processes. Due to uncertainty in the underlying dynamics of these phenomena, it is often challenging to determine how many modes should be used to describe the observed behaviors. By taking a Bayesian nonparametric approach, we are able to make fewer assumptions about the underlying dynamics than are required by a parametric approach, allowing the data to drive the complexity of the inferred model.

The most basic of the Markov switching processes we considered in this thesis is the hidden Markov model (HMM), which assumes that the data can be modeled as independent given an underlying discrete-valued Markov sequence. In Chapter 3, we examine a Bayesian nonparametric approach to learning HMMs with an unknown number of modes[1], and demonstrate that the current state of the art—the HDP-HMM—inadequately captures the temporal mode persistence present in our datasets of interest. We then show how one can augment the model with a learned bias towards self-transitions, resulting in what we refer to as the *sticky HDP-HMM.* One of the main contributions of this chapter is the formulation of this augmented model in a manner that integrates fully with Bayesian nonparametric inference. As a motivating example,

[1] Although Chapter 3 uses the terminology *state*, we switch to the *mode* terminology here to avoid confusion with the latent continuous-valued state sequence introduced in Chapter 4.

we consider the problem of *speaker diarization* in which the goal is to segment conference audio into a set of speaker labels in the presence of an unknown number of speakers. For this application, it is extremely unlikely that there are rapid transitions from speaker to speaker, and we show that capturing this temporal mode persistence is key in obtaining state-of-the-art speaker diarizations. Another key aspect of this application is the fact that the speaker-specific emissions are very complex and multimodal. This motivates our examination of a sticky HDP-HMM with Dirichlet process emissions; we show that the sticky parameter is essential in being able to identify such a model. In Chapter 3, we additionally present a blocked Gibbs sampler, relying on a truncated approximation to the sticky HDP-HMM, that leads to dramatic improvements in mixing rates over the previously proposed collapsed sequential Gibbs sampler.

Motivated by applications such as the dance of a honey bee, in Chapter 4 we develop Bayesian nonparametric approaches to learning more complex Markov switching processes in which each mode is endowed with conditionally linear dynamics (in contrast to the HMM's assumption of conditionally independent observations.) We consider two such Markov jump linear processes: the switching linear dynamic system (SLDS) and switching vector autoregressive (VAR) process. We refer to our Bayesian nonparametric versions of these models as the *HDP-SLDS* and *HDP-AR-HMM*, respectively. One of the challenges of these models is defining an appropriate prior on the dynamic parameters. We analyze a couple of possibilities including the conjugate matrix-normal inverse-Wishart prior (MNIW) and a sparsity-inducing automatic relevance determination (ARD) prior. Both of these priors require knowledge of the underlying model order (i.e., VAR order or latent state dimension); however, by employing the ARD prior so as to encourage sparsity in a very structured manner, we are able to make inferences about possible variable-order structure.

Finally, in Chapter 5 we turn to the case in which one has multiple related time series and would like to transfer knowledge among them. By jointly modeling the data, we aim to improve parameter estimates and find interesting relationships among the sequences. Following the theme of this thesis, we consider methods that allow each sequence to have an unknown number of dynamical behaviors. We show that simply tying together the parameters of multiple HDP-AR-HMMs is inadequate for our goals since such a model assumes each time-series is performing the same set of behaviors in the same manner. Instead of the global clustering induced by the Dirichlet process, we demonstrate that a featural representation induced by the beta process (referred to as the Indian buffet process or *IBP*) is more appropriate, encouraging sharing of behaviors among objects while still allowing time-series-specific variability. To perform inference on the resulting *IBP-AR-HMM*, we introduce a new method of sampling unique features in the non-conjugate IBP case based on a birth-death proposal. The overall MCMC sampler harnesses many efficiencies arising from the fact that based on a fixed set of features, the model reduces to a collection of finite switching VAR processes.

■ 6.2 Suggestions for Future Research

We conclude by discussing a variety of open research directions suggested by our approaches to Bayesian nonparametric learning of Markov switching processes. Each of the preceding chapters has presented a lengthy description of possible avenues for future research as the culminating section of the chapter. In the following we primarily point to important aspects from these sections, and highlight common themes that have appeared. We additionally present a few new concepts not previously discussed.

■ 6.2.1 Inference on Large-Scale Data

Split-Merge Proposals

In Sec. 3.6 and Sec. 5.5, we discuss a mixing rate issue common to both the sticky HDP-HMM and IBP-AR-HMM when examining high-dimensional datasets. Namely, if our samplers merge either two true HMM modes (in the case of the sticky HDP-HMM) or two true behavior categories (in the case of the IBP-AR-HMM), splitting this mode requires sampling a parameter from the high-dimensional prior distribution that better describes the data than currently instantiated parameters that have been informed by the data. Such occurrences are rare, leading to very slow mixing rates. Here, we clearly see the tradeoff between: marginalizing the model parameters and sequentially sampling the mode sequence, introducing strong temporal dependencies; and block-sampling the mode sequence, but requiring instantiation of parameters that must be sampled from high-dimensional priors. Clever initializations of the sampler (e.g., a block initialization that is expected to oversegment the mode sequence) can help with producing reasonable mode sequence samples by trapping the sampler within a preferred mode of the posterior, but still do not lead to fast mixing over the entire support of the posterior. In such cases, developing split-merge proposals similar to those developed for the Dirichlet process mixture model by Jain and Neal [77] could improve exploration between various modes of the posterior.

Variational Approaches

Although Gibbs sampling provides theoretical guarantees of accuracy, mixing rates on large datasets can often be slow, as described in the preceding section, and are in general difficult to characterize. Alternatively, variational inference [175] provides a fast, deterministic approximation to posteriors with an optimization criterion that can be easily utilized to assess convergence. The goal of the variational approach is to minimize the Kullbeck-Leibler (KL) divergence between the *variational distribution* and the posterior distribution with respect to the *free parameter.* This problem is generally intractable and is then "relaxed", yielding a simplified optimization problem that depends on multiple free parameters that are iteratively optimized. Variational schemes have been developed for the HDP [164] as well as linear dynamic systems [9]. However, the nonparametric nature of the dynamic models considered in this thesis raises new conceptual challenges in extending these previous works.

Online Learning

Another topic of keen interest in time series analysis is online estimation. The motivation for considering such algorithms is two-fold. First, and primarily, some applications require inferences to be made sequentially as data arrive. Another motivation arises from the fact that the batch processing algorithms developed in this thesis may be impractical for long time series datasets. Although the complexity of efficient algorithms like the forward-backward algorithm grow only linearly with the length of the dataset (versus quadratically with the dimension of the mode space), incorporating information from distant data may not be well motivated. Due both to the nonlinear dynamics and uncertainty in model parameters, exact recursive estimation is infeasible. As discussed in Sec. 4.4, we can instead leverage the *conditionally linear* dynamics to develop efficient sequential importance sampler techniques. The standard issue of a progressively impoverished particle representation, especially in the case of static parameter estimation, introduces challenges that are interesting to explore. See Sec. 4.4 for further details.

Lower Dimensional Analysis

For some of the applications we examine, it is possible that the dynamics of interest evolve on a lower-dimensional manifold. Methods for discovering such lower-dimensional descriptions of high-dimensional time-series is an open area of research. A first step at a lower-order description of the data would be to consider some dimensionality-reduction technique such as principal component analysis (PCA) [84]. We attempted PCA analysis of the speaker diarization data, but did not obtain promising results. Such techniques have been applied to motion capture data [7], and it would be interesting to see if such an approach would eliminate the step of hand-selecting gross motor measurements from the full 62-dimensional observation vector in the task presented in Sec. 5.4.

■ 6.2.2 Alternative Dynamic Structures

Semi-Markov Models

To address the temporal mode persistence issue in the HDP-HMM, we augmented the model with a bias on self-transitions. Although such an approach is reasonable and effective in many scenarios, maintaining the simplicity of the standard HMM structure, in some applications the dynamics are probably better described as semi-Markov [46]. That is, each mode is endowed with a duration distribution and the generative process dictates that upon entering a mode, a *sojourn time* within that mode is chosen from this duration distribution. At the culmination of the chosen sojourn time, the next mode is chosen from a transition distribution that solely depends upon the previous mode. When the duration distribution is chosen to be geometric, the semi-Markov formulation simplifies to that of a discrete-time Markov process. Various forms of hidden semi-Markov models (HSMM), where observations are of a latent semi-Markov mode sequence, have been proposed in the literature for finite mode spaces and are reviewed by Murphy [122]. Considering an HSMM with an unbounded mode space is a tangible extension of the HDP-HMM, made even more intriguing when considering each mode's duration distribution in a Bayesian nonparametric framework.

Time-Inhomogeneous Processes

The Markov switching processes we have considered in this thesis have assumed that the parameters associated with each mode, including the transition distributions and emission parameters, remain the same over time. An interesting direction for future research is to consider methods for relaxing this assumption and instead allowing for time-inhomogeneous processes. Take, for example, a person continually transitioning between an apriori unknown set of gaits (e.g., "run","walk","jog"). The parameters describing these gaits, and the relative frequency of each gait, may evolve in time. As alluded to in Sec. 5.5, one could develop extensions of the *dependent Dirchlet process* (DDP) [61, 111] to such tasks. Previous applications of the DDP include modeling changes in the firing of neurons [48] and document topic drift [156].

■ 6.2.3 Bayesian Nonparametric Variable-Order Markov Models

In Sec. 4.1.1 we explored a method for inferring variable-order switching VAR process by employing a sparsity-inducing prior on a fixed-dimensional parameter space. As mentioned in Sec. 4.4, the ARD prior we used simply provides a quadratic penalty on non-zero model parameters, and one could instead consider stronger sparsity-inducing priors like the class of spike and slab priors [28, 73, 182]. Such priors might lead to clearer analysis of non-dynamical model components since positive prior mass is placed on these components being exactly zero.

A possibly more intellectually gratifying direction for future research in this area is to consider a Bayesian nonparametric approach to learning variable-order models in which the maximal model order is unbounded. This idea is related to the infinite Markov model (iMM) [118] that uses prediction suffix trees (PST) to adaptively choose a Markov order, but with the significantly more challenging problem of continuous observations. A suffix tree stores strings of symbols at the nodes of a tree, and labels edges with the unique symbols. Each node of a PST is additionally associated with a distribution over the next symbol. A PST of depth n can be used for variable-order Markov modeling with a maximum order n [22]. Such models are often utilized in statistical language domains where contexts of varying length, such as "United States of", make certain words, such as "America", more likely. The iMM allows for infinite depth in an efficient manner by utilizing a stick-breaking process. However, this model relies on a discrete state space for repetition of observed strings. This will not occur, almost surely, with VAR processes. A promising first step is to consider variable-order moving average processes where we cluster on the input sequences and allow for additional noise. Extensions to VAR models is a challenging next step.

■ 6.2.4 Alternatives to Global Clustering

All of the models presented in this thesis (i.e., the sticky HDP-HMM of Chapter 3, the HDP-SLDS and HDP-AR-HMM of Chapter 4, and the IBP-AR-HMM of Chapter 5) assume that the association of an observation with a latent dynamical mode occurs based on every dimension of the observation vector. However, as we saw in the motion capture task of Sec. 5.4, some subjects shared behaviors solely in the trajectories of their lower or upper body. These differences resulted in splitting related behaviors into separate behavior categories. Motivated by this effect, instead of clustering based on *global* commonalities of dynamic parameters, one could consider clustering on *local* commonalities.

As discussed in Sec. 5.5, this idea is related to the *local partition process* (LPP) of [37]. The LPP assumes that the high-dimensional parameter vector describing one subject's collection of observations might be related to that of another subject, but only along certain dimensions of the parameter vector. A Dirichlet process prior on the parameter space would, on the other hand, imply that if two subjects share any component of the parameter vector, they share all components; this is unlikely to occur in high-dimensional applications. The formulation of [37] specifically considers a *random effects model* in the which the parameter vector is then referred to as a *random effects vector*. The LPP uses a combination of global and local clustering to create an effect where assigning two subjects to the same global cluster makes it more likely for the subjects to have identical values for multiple elements of their random effects vector. The closely related *kernel partition process* (KPP) [38] assumes that there is some natural ordering or location associated with the elements of the random effects vector, and incorporates the idea that if two subjects share a given component of their random effects vectors, they are more likely to share "close by" elements as well. It would be interesting to examine how to extend such processes to time-series models like the IBP-AR-HMM. For the motion capture data, one could assume that the coordinates of the parameter space are related via a distance metric based on the human body's geometry.

■ 6.2.5 Asymptotic Analysis

One element absent from this thesis is any sort of asymptotic analysis. As mentioned in Sec. 2.9.1, assuming the data are generated by a finite mixture, the Dirichlet process posterior is guaranteed to converge (in distribution) to that finite set of mixture parameters [75]. In addition, for target distributions with sufficiently small tail probabilities, Dirichlet process mixtures of Gaussians provide strongly consistent density estimates [54, 55]. Posterior consistency results also considering fat-tailed distributions like the Cauchy are presented in [168]. In [53], rates of convergence are established based on an assumption of a twice continuously differentiable target density. This paper also provides an overview of other convergence results based on various smoothness conditions.

The posterior consistency results above focused on density estimation using Dirichlet process mixtures of normals. In more general settings, the asymptotic behavior is more challenging to assess. Diaconis and Freedman [35] provide a scenario involving estimation of a Gaussian distributed location parameter based on independent, noisy observations where the noise distribution is unknown and given a Dirichlet process prior. Here, consistent frequentist estimates exist [86], but it is demonstrated that a heavy-tailed Student-t base measure may produce an inconsistent Bayes estimate. Other examples, such as the Robins-Ritov paradox [180], validate that care must be taken in analyzing the results produced by Bayesian nonparametric methods. Developing theoretical asymptotic guarantees for the Bayesian nonparametric models presented herein provides an important and challenging direction for future research.

Appendix A

Sticky HDP-HMM Direct Assignment Sampler

This appendix provides the derivations for the sequential, direct assignment Gibbs samplers outlined in Algorithms 9 and 11. Throughout this appendix, we will refer to the random variables in the graph of Fig. 2.13(b). For these derivations we include the κ term of the sticky HDP-HMM; the derivations for the original HDP-HMM follow directly by setting $\kappa = 0$.

■ A.1 Sticky HDP-HMM

To derive the direct assignment sampler for the sticky HDP-HMM, we first assume that we sample: table assignments for each customer, t_{ji}; served dish assignments for each table, k_{jt}; considered dish assignments, $\bar{k}_{jt}$; dish override variables, w_{jt}; and the global mixture weights, β. Because of the properties of the HDP, and more specifically the stick-breaking distribution, we are able to marginalize the group-specific distributions $\tilde{\pi}_j$ and parameters θ_k and still have closed-form distributions from which to sample (since exchangeability implies that we may treat every table and dish as if it were the last, as in Eq. (3.8).) The assumption of having t_{ji} and k_{jt} is a stronger assumption than that of having z_{ji} since z_{ji} can be uniquely determined from t_{ji} and k_{jt}, though not vice versa. We proceed to show that directly sampling z_{ji} instead of t_{ji} and k_{jt} is sufficient when the auxiliary variables m_{jk}, $\bar{m}_{jk}$, and w_{jt} are additionally sampled.

■ A.1.1 Sampling z_t

Recall the generative process of Eq. (3.1) using the sticky variant of the model given by Eq. (3.3). Using the conditional independence statements encoded in the graph of

Fig. 3.2(a), the posterior distribution of z_t factors as:

$$\begin{aligned} p(z_t = k \mid z_{\backslash t}, y_{1:T}, \beta, \alpha, \kappa) &\propto \int_{\boldsymbol{\pi}} \prod_i p(\pi_i \mid \alpha, \beta, \kappa) \prod_\tau p(z_\tau \mid \pi_{z_{\tau-1}}) d\boldsymbol{\pi} \\ &\quad \int \prod_k p(\theta_k \mid \lambda) \prod_\tau p(y_\tau \mid \theta_{z_\tau}) d\boldsymbol{\theta} \\ &\propto p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) p(y_t \mid y_{\backslash t}, z_t = k, z_{\backslash t}, \lambda). \qquad \text{(A.1)} \end{aligned}$$

The term $p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa)$, which arises from integration over $\boldsymbol{\pi}$, is a variant of the Chinese restaurant franchise prior, while $p(y_t \mid y_{\backslash t}, z_{t=k}, z_{\backslash t}, \lambda)$ is the likelihood of an assignment $z_t = k$ having marginalized the parameter θ_k.

The conditional distribution $p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa)$ of Eq. (A.1) can be written as:

$$\begin{aligned} p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) &\propto \int_{\boldsymbol{\pi}} p(z_{t+1} \mid \pi_k) p(z_t = k \mid \pi_{z_{t-1}}) \\ &\quad \prod_i (p(\pi_i \mid \alpha, \beta, \kappa) \prod_{\tau \mid z_{\tau-1} = i, \tau \neq t, t+1} p(z_\tau \mid \pi_i)) d\boldsymbol{\pi} \\ &\propto \int_{\boldsymbol{\pi}} p(z_{t+1} \mid \pi_k) p(z_t = k \mid \pi_{z_{t-1}}) \qquad \text{(A.2)} \\ &\quad \prod_i p(\pi_i \mid \{z_\tau \mid z_{\tau-1} = i, \tau \neq t, t+1\}, \beta, \alpha, \kappa) d\boldsymbol{\pi}. \end{aligned}$$

Let $z_{t-1} = j$. If $k \neq j$, that is, assuming a change in state value at time t, then

$$\begin{aligned} & p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) \\ &\quad \propto \int_{\pi_k} p(z_{t+1} \mid \pi_k) p(\pi_k \mid \{z_\tau \mid z_{\tau-1} = k, \tau \neq t, t+1\}, \beta, \alpha, \kappa) d\pi_k \\ &\quad \int_{\pi_j} p(z_t = k \mid \pi_j) p(\pi_j \mid \{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\}, \beta, \alpha, \kappa) d\pi_j \\ &\quad \propto p(z_{t+1} \mid \{z_\tau \mid z_{\tau-1} = k, \tau \neq t, t+1\}, \beta, \alpha, \kappa) \qquad \text{(A.3)} \\ &\quad\quad p(z_t = k \mid \{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\}, \beta, \alpha, \kappa). \end{aligned}$$

When considering the probability of a self-transition (i.e., $k = j$), we have

$$\begin{aligned} p(z_t = j \mid z_{\backslash t}, \beta, \alpha, \kappa) &\propto \int_{\pi_j} p(z_{t+1} \mid \pi_j) p(z_t = j \mid \pi_j) \\ &\quad p(\pi_j \mid \{z_\tau \mid z_{\tau-1} = k, \tau \neq t, t+1\}, \beta, \alpha, \kappa) d\pi_j \\ &\propto p(z_t = j, z_{t+1} \mid \{z_\tau \mid z_{\tau-1} = k, \tau \neq t, t+1\}, \beta, \alpha, \kappa). \qquad \text{(A.4)} \end{aligned}$$

These predictive distributions can be derived by standard results arising from having placed a Dirichlet prior on the parameters defining these multinomial observations z_τ.

The finite Dirichlet prior is induced by considering the finite partition $\{1, \dots, K, A_{\tilde{k}}\}$ of $\mathbb{Z}_+$, where $A_{\tilde{k}} = \{K+1, K+2, \dots\}$ is the set of unrepresented state values in $z_{\backslash t}$. The properties of the Dirichlet process (see Theorem 2.9.1) dictate that on this finite partition, we have the following form for the group-specific transition distributions:

$$\pi_j \mid \alpha, \beta \sim \mathrm{Dir}(\alpha\beta_1, \dots, \alpha\beta_j + \kappa, \dots, \alpha\beta_K, \alpha\beta_{\tilde{k}}), \tag{A.5}$$

where $\beta_{\tilde{k}} = \sum_{i=K+1}^{\infty} \beta_i$. Using this prior, we derive the distribution of a generic set of observations generated from a single transition distribution π_i given the hyperparameters α, β, and κ:

$$\begin{aligned}
p(\{z_\tau \mid z_{\tau-1} = i\} \mid \beta, \alpha, \kappa) &= \int_{\pi_i} p(\pi_i \mid \beta, \alpha, \kappa) p(\{z_\tau \mid z_{\tau-1} = i\} \mid \pi_i) d\pi_i \\
&= \int_{\pi_i} \frac{\Gamma(\sum_k \alpha\beta_k + \kappa\delta(k,i))}{\prod_k \Gamma(\alpha\beta_k + \kappa\delta(k,i))} \prod_{k=1}^{K+1} \pi_{jk}^{\alpha\beta_k + \kappa\delta(k,i) - 1} \prod_{k=1}^{K+1} \pi_{jk}^{n_{jk}} d\pi_i \\
&= \frac{\Gamma(\sum_k \alpha\beta_k + \kappa\delta(k,i))}{\prod_k \Gamma(\alpha\beta_k + \kappa\delta(k,i))} \frac{\prod_k \Gamma(\alpha\beta_k + \kappa\delta(k,i) + n_{jk})}{\Gamma(\sum_k \alpha\beta_k + \kappa\delta(k,i) + n_{jk})} \\
&= \frac{\Gamma(\alpha + \kappa)}{\Gamma(\alpha + \kappa + n_{i\cdot})} \prod_k \frac{\Gamma(\alpha\beta_k + \kappa\delta(k,i) + n_{jk})}{\Gamma(\alpha\beta_k + \kappa\delta(k,i))},
\end{aligned} \tag{A.6}$$

where we make a slight abuse of notation in taking $\beta_{K+1} = \beta_{\tilde{k}}$. We use Eq. (A.6) to determine that the first component of Eq. (A.3) is

$$\begin{aligned}
&p(z_t = k \mid \{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\}, \beta, \alpha, \kappa) \\
&\qquad = \frac{p(\{z_\tau \mid z_{\tau-1} = j, \tau \neq t+1, z_t = k\} \mid \beta, \alpha, \kappa)}{p(\{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\} \mid \beta, \alpha, \kappa)} \\
&\qquad = \frac{\Gamma(\alpha + \kappa + n_{j\cdot}^{-t})}{\Gamma(\alpha + n_{j\cdot}^{-t} + 1)} \frac{\Gamma(\alpha\beta_k + \kappa + n_{jk}^{-t} + 1)}{\Gamma(\alpha\beta_k + n_{jk}^{-t})} = \frac{\alpha\beta_k + n_{jk}^{-t}}{\alpha + n_{j\cdot}^{-t}}.
\end{aligned} \tag{A.7}$$

Here, n_{jk}^{-t} denotes the number of transitions from state j to k not counting the transition from z_{t-1} to z_t or from z_t to z_{t+1}. Similarly, the second component of Eq. (A.3) is derived to be

$$p(z_{t+1} = \ell \mid \{z_\tau \mid z_{\tau-1} = k, \tau \neq t, t+1\}, \beta, \alpha, \kappa) = \frac{\alpha\beta_\ell + \kappa\delta(\ell, k) + n_{k\ell}^{-t}}{\alpha + \kappa + n_{k\cdot}^{-t}}, \tag{A.8}$$

For $k = j$, the distribution of Eq. (A.4) reduces to

$$\begin{aligned}
&p(z_t = j, z_{t+1} \mid \{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\}, \beta, \alpha, \kappa) \\
&\quad = \frac{p(\{z_\tau \mid z_{\tau-1} = j\} \mid \beta, \alpha, \kappa)}{p(\{z_\tau \mid z_{\tau-1} = j, \tau \neq t, t+1\} \mid \beta, \alpha, \kappa)} \\
&\quad = \begin{cases} \frac{\Gamma(\alpha+\kappa+n_{j\cdot}^{-t})}{\Gamma(\alpha+\kappa+n_{j\cdot}^{-t}+2)} \frac{\Gamma(\alpha\beta_j+\kappa+n_{jj}^{-t}+1)}{\Gamma(\alpha\beta_j+\kappa+n_{jj}^{-t})} \frac{\Gamma(\alpha\beta_\ell+n_{j\ell}^{-t}+1)}{\Gamma(\alpha\beta_\ell+n_{j\ell}^{-t})}, & z_{t+1} = \ell, \ell \neq j; \\ \frac{\Gamma(\alpha+\kappa+n_{j\cdot}^{-t})}{\Gamma(\alpha+\kappa+n_{j\cdot}^{-t}+2)} \frac{\Gamma(\alpha\beta_j+\kappa+n_{jj}^{-t}+2)}{\Gamma(\alpha\beta_j+\kappa+n_{jj}^{-t})}, & z_{t+1} = j; \end{cases} \\
&\quad = \begin{cases} \frac{(\alpha\beta_j+\kappa+n_{jj}^{-t})(\alpha\beta_\ell+n_{j\ell}^{-t})}{(\alpha+\kappa+n_{j\cdot}^{-t}+1)(\alpha+\kappa+n_{j\cdot}^{-t})}, & z_{t+1} = \ell, \ell \neq j; \\ \frac{(\alpha\beta_j+\kappa+n_{jj}^{-t}+1)(\alpha\beta_j+\kappa+n_{jj}^{-t})}{(\alpha+\kappa+n_{j\cdot}^{-t}+1)(\alpha+\kappa+n_{j\cdot}^{-t})}, & z_{t+1} = j; \end{cases} \\
&\quad = \frac{(\alpha\beta_j + \kappa + n_{jj}^{-t})(\alpha\beta_\ell + n_{j\ell}^{-t} + (\kappa+1)\delta(j,\ell))}{(\alpha + \kappa + n_{j\cdot}^{-t})(\alpha + \kappa + n_{j\cdot}^{-t} + 1)}. \qquad \text{(A.9)}
\end{aligned}$$

Combining these cases, the prior predictive distribution of z_t is:

$$\begin{aligned}
&p(z_t = k \mid z_{\setminus t}, \beta, \alpha, \kappa) \\
&\quad \propto \begin{cases} (\alpha\beta_k + n_{z_{t-1}k}^{-t} + \kappa\delta(z_{t-1}, k)) \\ \qquad \left(\frac{\alpha\beta_{z_{t+1}} + n_{kz_{t+1}}^{-t} + \kappa\delta(k, z_{t+1}) + \delta(z_{t-1}, k)\delta(k, z_{t+1})}{\alpha + n_{k\cdot}^{-t} + \kappa + \delta(z_{t-1}, k)} \right) & k \in \{1, \dots, K\} \\ \frac{\alpha^2 \beta_{\tilde{k}} \beta_{z_{t+1}}}{\alpha+\kappa} & k = K+1. \end{cases} \qquad \text{(A.10)}
\end{aligned}$$

The conditional distribution of the observation y_t given an assignment $z_t = k$ and given all other observations y_τ, having marginalized out θ_k, can be written as follows:

$$\begin{aligned}
p(y_t \mid y_{\setminus t}, z_t = k, z_{\setminus t}, \lambda) &\propto \int p(y_t \mid \theta_k) p(\theta_k \mid \lambda) \prod_{\tau \mid z_\tau = k, \tau \neq t} p(y_\tau \mid \theta_k) d\theta_k \\
&\propto \int p(y_t \mid \theta_k) p(\theta_k \mid \{y_\tau \mid z_\tau = k, \tau \neq t\}, \lambda) d\theta_k \\
&\propto p(y_t \mid \{y_\tau \mid z_\tau = k, \tau \neq t\}, \lambda). \qquad \text{(A.11)}
\end{aligned}$$

There exists a closed-form distribution for this likelihood if we consider a conjugate distribution on the parameter space Θ.

Assuming our emission distributions are Gaussian with unknown mean and covariance parameters, the conjugate prior is the normal-inverse-Wishart distribution (see Sec. 2.4.3), which we denote by $\mathcal{NIW}(\zeta, \boldsymbol{\vartheta}, \nu, \Delta)$. Here, $\lambda = \{\zeta, \boldsymbol{\vartheta}, \nu, \Delta\}$. As described in Sec. 2.4.3, via conjugacy, the posterior distribution of $\theta_k = \{\mu_k, \Sigma_k\}$ given a set of Gaussian observations $y_t \sim \mathcal{N}(\mu_k, \Sigma_k)$ is distributed as an updated normal-inverse-

Wishart $\mathcal{NIW}(\bar{\zeta}_k, \bar{\boldsymbol{\vartheta}}_k, \bar{\nu}_k, \bar{\Delta}_k)$, where

$$\begin{aligned}
\bar{\zeta}_k &= \zeta + |\{y_s \mid z_s = k, s \neq t\}| \triangleq \zeta + |Y_k| \\
\bar{\nu}_k &= \nu + |Y_k| \\
\bar{\zeta}_k \bar{\boldsymbol{\vartheta}}_k &= \zeta \boldsymbol{\vartheta} + \sum_{y_s \in Y_k} y_s \\
\bar{\nu}_k \bar{\Delta}_k &= \nu \Delta + \sum_{y_s \in Y_k} y_s y_s^T + \zeta \boldsymbol{\vartheta} \boldsymbol{\vartheta}^T - \bar{\zeta}_k \bar{\boldsymbol{\vartheta}}_k \bar{\boldsymbol{\vartheta}}_k^T.
\end{aligned}$$

Marginalizing θ_k induces a multivariate Student-t predictive distribution for y_t as given by Eq. (2.87):

$$\begin{aligned}
p(y_t \mid \{y_\tau \mid z_\tau = k, \tau \neq t\}, \zeta, \boldsymbol{\vartheta}, \nu, \Delta) &= t_{\bar{\nu}_k - d - 1}\left(y_t; \bar{\boldsymbol{\vartheta}}_k, \frac{(\bar{\zeta}_k + 1)\bar{\nu}_k}{\bar{\zeta}_k(\bar{\nu}_k - d - 1)} \bar{\Delta}_k\right) \\
&\triangleq t_{\hat{\nu}_k}(y_t; \hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k). \qquad \text{(A.12)}
\end{aligned}$$

■ A.1.2 Sampling β

Let $\bar{K}$ be the number of unique dishes *considered.* We note that for the sticky HDP-HMM, every served dish had to be considered in some restaurant. The only scenario in which this would not be the case is if for some dish j, every table served dish j arose from an override decision. However, overrides resulting in dish j being served can only occur in restaurant j, and this restaurant would not exist if dish j was not considered (and thus served) in some other restaurant. Therefore, each served dish had to be considered by at least one table in the franchise. On the other hand, there may be some dishes considered that were never served. From this, we conclude that $\bar{K} \geq K$. We will assume that the K served dishes are indexed in $\{1, \dots, K\}$ and any considered, but not served, dish is indexed in $\{K+1, K+2, \dots\}$. For the sake of inference, we will see in the following section that $\bar{K}$ never exceeds K, the number of unique considered dishes, implying that $\bar{K} = K$.

Take a finite partition $\{\theta_1, \theta_2, \dots, \theta_{\bar{K}}, \Theta_{\tilde{k}}\}$ of the parameter space Θ, where $\Theta_{\tilde{k}} = \Theta \backslash \bigcup_{k=1}^{\bar{K}} \{\theta_k\}$ is the set of all currently unrepresented parameters. By definition of the Dirichlet process (once again using Theorem 2.9.1 combined with the fact that $G_0 \sim \mathrm{DP}(\gamma, H)$), G_0 has the following distribution on this finite partition:

$$\begin{aligned}
(G_0(\theta_1), \dots, G_0(\theta_{\bar{K}}), G_0(\Theta_{\tilde{k}})) \mid \gamma, H &\sim \mathrm{Dir}(\gamma H(\theta_1), \dots, \gamma H(\theta_{\bar{K}}), \gamma H(\Theta_{\tilde{k}})) \\
&\sim \mathrm{Dir}(0, \dots, 0, \gamma), \qquad \text{(A.13)}
\end{aligned}$$

where we have used the fact that H is absolutely continuous with respect to Lebesgue measure.

For every currently instantiated table t, the considered dish assignment variable $\bar{k}_{jt}$ associates the table-specific considered dish θ_{jt}^* with one among the unique set of dishes $\{\theta_1, \dots, \theta_{\bar{K}}\}$. Recalling that $\bar{m}_{jk}$ denotes how many of the tables in restaurant j

considered dish θ_k, we see that we have $\bar{m}_{\cdot k}$ observations $\theta^*_{jt} \sim G_0$ in the franchise that fall within the single-element cell $\{\theta_k\}$. By the properties of the Dirichlet distribution, specifically as given by Eq. (2.74), the posterior of G_0 is

$$(G_0(\theta_1), \dots, G_0(\theta_{\bar{K}}), G_0(\Theta_{\tilde{k}}))|\boldsymbol{\theta}^*, \gamma \sim \mathrm{Dir}(\bar{m}_{\cdot 1}, \dots, \bar{m}_{\cdot \bar{K}}, \gamma). \tag{A.14}$$

Since $(G_0(\theta_1), \dots, G_0(\theta_{\bar{K}}), G_0(\Theta_{\tilde{k}}))$ is by definition equal to $(\beta_1, \dots, \beta_{\bar{K}}, \beta_{\tilde{k}})$, and from the conditional independencies illustrated in Fig. 2.13, the desired posterior of β is

$$(\beta_1, \dots, \beta_{\bar{K}}, \beta_{\tilde{k}}) \mid \boldsymbol{t}, \boldsymbol{k}, \bar{\boldsymbol{k}}, \boldsymbol{w}, y_{1:T}, \gamma \sim \mathrm{Dir}(\bar{m}_{\cdot 1}, \dots, \bar{m}_{\cdot \bar{K}}, \gamma), \tag{A.15}$$

where here we define $\beta_{\tilde{k}} = \sum_{k=\bar{K}+1}^{\infty} \beta_k$. From the above, we see that $\{\bar{m}_{\cdot k}\}_{k=1}^{\bar{K}}$ is a set of sufficient statistics for resampling β defined on this partition. Thus, it is sufficient to sample $\bar{m}_{jk}$ instead of t_{ji} and k_{jt}, when given the state index z_t. The sampling of $\bar{m}_{jk}$, as well as the resampling of hyperparameters (see Appendix C), is greatly simplified by additionally sampling auxiliary variables m_{jk} and w_{jt}, corresponding to the number of tables in restaurant j that were *served* dish k and the corresponding override variables.

A.1.3 Jointly Sampling m_{jk}, w_{jt}, and $\bar{m}_{jk}$

We jointly sample the auxiliary variables m_{jk}, w_{jt}, and $\bar{m}_{jk}$ from

$$\begin{aligned} p(\boldsymbol{m}, \boldsymbol{w}, \bar{\boldsymbol{m}} \mid z_{1:T}, \beta, \alpha, \kappa) = p(\bar{\boldsymbol{m}} \mid \boldsymbol{m}, \boldsymbol{w}, z_{1:T}, \beta, \alpha, \kappa)& \\ p(\boldsymbol{w} \mid \boldsymbol{m}, z_{1:T}, \beta, \alpha, \kappa) p(\boldsymbol{m} \mid z_{1:T}, \beta, \alpha, \kappa).& \end{aligned} \tag{A.16}$$

We start by examining $p(\boldsymbol{m} \mid z_{1:T}, \beta, \alpha, \kappa)$. Having the state index assignments $z_{1:T}$ effectively partitions the data (customers) into both restaurants and dishes, though the table assignments are unknown since multiple tables can be served the same dish. Thus, sampling m_{jk} is in effect equivalent to sampling table assignments for each customer *after* knowing the dish assignment. This conditional distribution is given by:

$$\begin{aligned} p(t_{ji} = t \mid k_{jt} = k, \boldsymbol{t}^{-ji}, \boldsymbol{k}^{-jt}, y_{1:T}, \beta, \alpha, \kappa)& \\ \propto p(t_{ji} \mid t_{j1}, \dots, t_{ji-1}, t_{ji+1}, \dots, t_{jT_j}, \alpha, \kappa) p(k_{jt} = k \mid \beta, \alpha, \kappa)& \\ \propto \begin{cases} \tilde{n}_{jt}^{-ji}, & t \in \{1, \dots, T_j\}; \\ \alpha\beta_k + \kappa\delta(k, j), & t = T_j + 1, \end{cases}& \end{aligned} \tag{A.17}$$

where $\tilde{n}_{jt}^{-ji}$ is the number of customers sitting at table t in restaurant j, not counting y_{ji}. Similarly, $\boldsymbol{t}^{-ji}$ are the table assignments for all customers except y_{ji} and $\boldsymbol{k}^{-jt}$ are the dish assignments for all tables except table t in restaurant j. We recall from Sec. 3.1.1 that T_j is the number of currently occupied tables in restaurant j. The form of Eq. (A.17) implies that a customer's table assignment conditioned on a dish assignment k follows a Dirichlet process with concentration parameter $\alpha\beta_k + \kappa\delta(k, j)$. That is,

$$t_{ji} \mid k_{jt_{ji}} = k, \boldsymbol{t}^{-ji}, \boldsymbol{k}^{-jt_{ji}}, y_{1:T}, \beta, \alpha, \kappa \sim \tilde{\pi}', \qquad \tilde{\pi}' \sim \mathrm{GEM}(\alpha\beta_k + \kappa\delta(k, j)). \tag{A.18}$$

Then, Eq. (2.218) provides the form for the distribution over the number of unique components (i.e., tables) generated by sampling n_{jk} times from this stick-breaking distributed measure, where we note that for the HDP-HMM n_{jk} is the number of customers in restaurant j eating dish k:

$$p(m_{jk} = m \mid n_{jk}, \beta, \alpha, \kappa) = \frac{\Gamma(\alpha\beta_k + \kappa\delta(k,j))}{\Gamma(\alpha\beta_k + \kappa\delta(k,j) + n_{jk})} s(n_{jk}, m)(\alpha\beta_k + \kappa\delta(k,j))^m. \tag{A.19}$$

For large n_{jk}, it is often more efficient to sample m_{jk} by simulating the table assignments of the Chinese restaurant, as described by Eq. (A.17), rather than having to compute a large array of Stirling numbers.

We now derive the conditional distribution for the override variables w_{jt}. The table counts provide that m_{jk} tables are serving dish k in restaurant j. If $k \neq j$, we automatically have m_{jk} tables with $w_{jt} = 0$ since the served dish is not the house specialty. Otherwise, for each of the m_{jj} tables t serving dish $k_{jt} = j$, we start by assuming we know the considered dish index $\bar{k}_{jt}$, from which inference of the override parameter is trivial. We then marginalize over all possible values of this index:

$$\begin{aligned} p(w_{jt} \mid k_{jt} = j, \beta, \rho = &\sum_{\bar{k}_{jt}=1}^{\bar{K}} p(\bar{k}_{jt}, w_{jt} \mid k_{jt} = j, \beta) + p(\bar{k}_{jt} = \bar{K}+1, w_{jt} \mid k_{jt} = j, \beta) \\ &\propto \sum_{\bar{k}_{jt}=1}^{\bar{K}} p(k_{jt} = j \mid \bar{k}_{jt}, w_{jt}) p(\bar{k}_{jt} \mid \beta) p(w_{jt} \mid \rho) \\ &\quad + p(k_{jt} = j \mid \bar{k}_{jt} = \bar{K}+1, w_{jt}) p(\bar{k}_{jt} = \bar{K}+1 \mid \beta) p(w_{jt} \mid \rho) \\ &\propto \begin{cases} \beta_j(1-\rho), & w_{jt} = 0; \\ \rho, & w_{jt} = 1, \end{cases} \end{aligned} \tag{A.20}$$

where $\rho = \frac{\kappa}{\alpha+\kappa}$ is the prior probability that $w_{jt} = 1$. This distribution implies that having observed a served dish $k_{jt} = j$ makes it more likely that the considered dish $\bar{k}_{jt}$ was overridden via choosing $w_{jt} = 1$ than the prior suggests. This is justified by the fact that if $w_{jt} = 1$, the considered dish $\bar{k}_{jt}$ could have taken any value and the served dish would still be $k_{jt} = j$. The only other explanation of the observation $k_{jt} = j$ is that the dish was not overridden, namely $w_{jt} = 0$ occurring with prior probability $(1 - \rho)$, *and* the table considered a dish $\bar{k}_{jt} = j$, occurring with probability β_j. These events are independent, resulting in the above distribution. We draw m_{jj} i.i.d. samples of w_{jt} from Eq. (A.20), with the total number of dish overrides in restaurant j given by $w_{j\cdot} = \sum_t w_{jt}$. The sum of these Bernoulli random variables results in a binomial random variable.

Given m_{jk} for all j and k and w_{jt} for each of these instantiated tables, we can now deterministically compute $\bar{m}_{jk}$, the number of tables that *considered* ordering dish k

in restaurant j. Any table that was overridden is an uninformative observation for the posterior of $\bar{m}_{jk}$ so that

$$\bar{m}_{jk} = \begin{cases} m_{jk}, & j \neq k; \\ m_{jj} - w_{j\cdot}, & j = k. \end{cases} \tag{A.21}$$

Note that we are able to subtract off the sum of the override variables within a restaurant, $w_{j\cdot}$, since the only time $w_{jt} = 1$ is if table t is served dish j. From Eq. (A.21), we see that $\bar{K} = K$.

■ A.2 Sticky HDP-HMM with DP emissions

In this section we derive the predictive distribution of the augmented state (z_t, s_t) of the sticky HDP-HMM with DP emissions of Sec. 3.3. We use the chain rule to write:

$$\begin{aligned} p(z_t = k, s_t = j \mid z_{\backslash t}, s_{\backslash t}, y_{1:T}, \beta, \alpha, \sigma, \kappa, \lambda) \\ = p(s_t = j \mid z_t = k, z_{\backslash t}, s_{\backslash t}, y_{1:T}, \sigma, \lambda) \\ p(z_t = k \mid z_{\backslash t}, s_{\backslash t}, y_{1:T}, \beta, \alpha, \kappa, \lambda). \end{aligned} \tag{A.22}$$

We can examine each term of this distribution by once again considering the joint distribution over all variables in the graph of Fig. 3.2(b) and integrating over the appropriate parameters. For the conditional distribution of $z_t = k$ when *not* given s_t, this amounts to:

$$\begin{aligned} p(z_t = k \mid z_{\backslash t}, s_{\backslash t}, y_{1:T}, \beta, \alpha, \kappa, \lambda) &\propto \int_{\boldsymbol{\pi}} \prod_j p(\pi_j \mid \alpha, \beta, \kappa) \prod_\tau p(z_\tau \mid \pi_{z_{\tau-1}}) d\boldsymbol{\pi} \\ &\quad \sum_{s_t} \int_{\boldsymbol{\psi}} \prod_j p(\psi_j \mid \sigma) \prod_\tau p(s_\tau \mid \psi_{z_\tau}) d\boldsymbol{\psi} \int \prod_{i,\ell} p(\theta_{i,\ell} \mid \lambda) \prod_\tau p(y_\tau \mid \theta_{z_\tau, s_\tau}) d\boldsymbol{\theta} \\ &\propto p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) \\ &\quad \sum_{s_t} p(s_t \mid \{s_\tau \mid z_\tau = k, \tau \neq t\}, \sigma) p(y_t \mid \{y_\tau \mid z_\tau = k, s_t, \tau \neq t\}, \lambda). \end{aligned} \tag{A.23}$$

The term $p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa)$ is as in Eq. (A.10), while

$$p(s_t = j \mid \{s_\tau \mid z_\tau = k, \tau \neq t\}, \sigma) = \begin{cases} \frac{n_{kj}^{\prime -t}}{\sigma + n_{k\cdot}^{\prime -t}}, & j \in \{1, \ldots, K_k^\prime\}; \\ \frac{\sigma}{\sigma + n_{k\cdot}^{\prime -t}}, & j = K_k^\prime + 1, \end{cases} \tag{A.24}$$

which is the predictive distribution of the indicator random variables of the DP mixture model associated with $z_t = k$. This can be derived directly from the Chinese restaurant process predictive distribution of Eq. (2.216). Here, $n_{kj}^{\prime -t}$ is the number of observations

y_τ with $(z_\tau = k, s_\tau = j)$ for $\tau \neq t$, and K'_k is the number of currently instantiated mixture components for the k^{th} emission density.

We similarly derive the conditional distribution of an assignment $s_t = j$ given $z_t = k$ as:

$$\begin{aligned} p(s_t = j \mid z_t = k, z_{\backslash t}, s_{\backslash t}, y_{1:T}, \sigma, \lambda) \propto \; & p(s_t = j \mid \{s_\tau \mid z_\tau = k, \tau \neq t\}, \sigma) \\ & p(y_t \mid \{y_\tau \mid z_\tau = k, s_t = j, \tau \neq t\}, \lambda). \end{aligned} \tag{A.25}$$

The likelihood component of these distributions,

$$p(y_t \mid \{y_\tau \mid z_\tau = k, s_t = j, \tau \neq t\}, \lambda), \tag{A.26}$$

is derived in the same fashion as Eq. (A.12) where now we only consider the observations y_τ that are assigned to HDP-HMM state $z_\tau = k$ *and* mixture component $s_\tau = k$.

Appendix B

Sticky HDP-HMM Blocked Sampler

In this appendix, we present the derivation of the blocked Gibbs samplers outlined in Algorithms 10 and 12.

B.1 Sampling β, π, and ψ

The order L weak limit approximation to the Dirichlet process mixture model gives us the following form for the prior distribution on the global weights β (see Eq. (2.225)):

$$\beta \mid \gamma \sim \text{Dir}(\gamma/L, \dots, \gamma/L). \tag{B.1}$$

Using the sticky HPD-HMM generative model of Eq. (3.3), Theorem 2.9.1 informs us that on this finite partition, the prior distribution over the transition distributions is Dirichlet with parametrization:

$$\pi_j \mid \alpha, \kappa, \beta \sim \text{Dir}(\alpha\beta_1, \dots, \alpha\beta_j + \kappa, \dots, \alpha\beta_L). \tag{B.2}$$

Recalling that $\bar{k}_{jt} \sim \beta$ and $z_t \sim \pi_{z_{t-1}}$, the standard Dirichlet-multinomial conjugacy results of Eq. (2.74) imply that the posterior distributions are given by:

$$\begin{aligned} \beta \mid \bar{\boldsymbol{m}}, \gamma &\sim \text{Dir}(\gamma/L + \bar{m}_{\cdot 1}, \dots, \gamma/L + \bar{m}_{\cdot L}) \\ \pi_j \mid z_{1:T}, \alpha, \beta &\sim \text{Dir}(\alpha\beta_1 + n_{j1}, \dots, \alpha\beta_j + \kappa + n_{jj}, \dots, \alpha\beta_L + n_{jL}), \end{aligned} \tag{B.3}$$

where we recall that n_{jk} is the number of j to k transitions in the state sequence $z_{1:T}$ and $\bar{m}_{jk}$ is the number of tables in restaurant j that considered dish k. The sampling of the auxiliary variables $\bar{m}_{jk}$ is as in Appendix A.

For the sticky HDP-HMM with DP emissions of Sec. 3.3, an order L' weak limit approximation to the DP prior on the emission parameters yields the following posterior distribution on the mixture weights ψ_k:

$$\psi_k \mid z_{1:T}, s_{1:T}, \sigma \sim \text{Dir}(\sigma/L' + n'_{k1}, \dots, \sigma/L' + n'_{kL'}), \tag{B.4}$$

where $n'_{k\ell}$ is the number of observations assigned to the ℓ^{th} mixture component of the k^{th} HMM state.

B.2 Sampling $z_{1:T}$ for the Sticky HDP-HMM

To derive the forward-backward procedure for jointly sampling $z_{1:T}$ given $y_{1:T}$ for the sticky HDP-HMM, we first note that

$$p(z_{1:T} \mid y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(z_T \mid z_{T-1}, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_{T-1} \mid z_{T-2}, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ \cdots p(z_2 \mid z_1, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_1 \mid y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}).$$

Thus, we may first sample z_1 from $p(z_1 \mid y_{1:T}, \boldsymbol{\pi}, \beta, \boldsymbol{\theta})$, then condition on this value to sample z_2 from $p(z_2 \mid z_1, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta})$, and so on. The conditional distribution of z_1 is derived as:

$$\begin{aligned} p(z_1 \mid y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) &\propto p(z_1) p(y_1 \mid \theta_{z_1}) \sum_{z_{2:T}} \prod_t p(z_t \mid \pi_{z_{t-1}}) p(y_t \mid \theta_{z_t}) \\ &\propto p(z_1) p(y_1 \mid \theta_{z_1}) \sum_{z_2} p(z_2 \mid \pi_{z_1}) p(y_2 \mid \theta_{z_2}) m_{3,2}(z_2) \\ &\propto p(z_1) p(y_1 \mid \theta_{z_1}) m_{2,1}(z_1), \end{aligned} \tag{B.5}$$

where $m_{t,t-1}(z_{t-1})$ is the backward message passed from z_t to z_{t-1} and for an HMM is recursively defined by (see Sec. 2.6.1):

$$\begin{aligned} m_{t,t-1}(z_{t-1}) &\propto \begin{cases} \sum_{z_t} p(z_t \mid \pi_{z_{t-1}}) p(y_t \mid \theta_{z_t}) m_{t+1,t}(z_t), & t \le T; \\ 1, & t = T+1; \end{cases} \\ &\propto p(y_{t:T} \mid z_{t-1}, \boldsymbol{\pi}, \boldsymbol{\theta}). \end{aligned} \tag{B.6}$$

The general conditional distribution of z_t is:

$$p(z_t \mid z_{t-1}, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto p(z_t \mid \pi_{z_{t-1}}) p(y_t \mid \theta_{z_t}) m_{t+1,t}(z_t). \tag{B.7}$$

So, to block sample $z_{1:T}$, we pass messages backwards and then recursively sample z_t forwards (i.e., for $t = 1, \ldots, T$) from the distributions defined in Eq. (B.7) and Eq. (B.5).

B.3 Sampling $(z_{1:T}, s_{1:T})$ for the Sticky HDP-HMM with DP emissions

We now examine how to sample the augmented state (z_t, s_t) of the sticky HDP-HMM with DP emissions. The conditional distribution of (z_t, s_t) for the forward-backward procedure is derived as:

$$p(z_t, s_t \mid z_{t-1}, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta}) \propto p(z_t \mid \pi_{z_{t-1}}) p(s_t \mid \psi_{z_t}) p(y_t \mid \theta_{z_t, s_t}) m_{t+1,t}(z_t). \tag{B.8}$$

Since the Markovian structure is only on the z_t component of the augmented state, the backward message $m_{t,t-1}(z_{t-1})$ from (z_t, s_t) to (z_{t-1}, s_{t-1}) is solely a function of z_{t-1}. These messages are given by (see Sec. 2.6.1):

$$\begin{aligned} &m_{t,t-1}(z_{t-1}) \\ &\quad \propto \begin{cases} \sum_{z_t} \sum_{s_t} p(z_t \mid \pi_{z_{t-1}}) p(s_t \mid \psi_{z_t}) p(y_t \mid \theta_{z_t, s_t}) m_{t+1,t}(z_t), & t \le T; \\ 1, & t = T+1. \end{cases} \end{aligned} \tag{B.9}$$

More specifically, since each component j of the k^{th} state-specific emission distribution is a Gaussian with parameters $\theta_{j,k} = \{\mu_{k,j}, \Sigma_{k,j}\}$, we have:

$$\begin{aligned} p(z_t = k, s_t = j \mid z_{t-1}, y_{1:T}, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta}) &\propto \pi_{z_{t-1}}(k)\psi_k(j)\mathcal{N}(y_t; \mu_{k,j}, \Sigma_{k,j})m_{t+1,t}(k) \\ m_{t+1,t}(k) &= \sum_{i=1}^{L}\sum_{\ell=1}^{L'} \pi_k(i)\psi_i(\ell)\mathcal{N}(y_{t+1}; \mu_{i,\ell}, \Sigma_{i,\ell})m_{t+2,t+1}(i) \\ m_{T+1,T}(k) &= 1 \quad k = 1, \dots, L. \end{aligned} \tag{B.10}$$

B.4 Sampling θ

Depending on the form of the emission distribution and base measure on the parameter space Θ, we sample parameters for each of the currently instantiated states from the updated posterior distribution. For the sticky HDP-HMM, this distribution is:

$$\theta_j \mid z_{1:T}, y_{1:T}, \lambda \sim p(\theta \mid \{y_t \mid z_t = j\}, \lambda). \tag{B.11}$$

For the sticky HDP-HMM with DP emissions, the posterior distribution for each Gaussian's mean and covariance, $\theta_{k,j}$, is determined by the observations assigned to this component, namely,

$$\theta_{k,j} \mid z_{1:T}, s_{1:T}, y_{1:T}, \lambda \sim p(\theta \mid \{y_t \mid (z_t = k, s_t = j)\}, \lambda). \tag{B.12}$$

B.4.1 Non-Conjugate Base Measures

Since the blocked sampler instantiates the parameters θ_k, rather than marginalizing them as in the direct assignment sampler, we can place a non-conjugate base measure on the parameter space Θ. Take, for example, the case of single Gaussian emission distributions where the parameters are the means and covariances of these distributions. Here, $\theta_k = \{\mu_k, \Sigma_k\}$. In this situation, one may place a Gaussian prior $\mathcal{N}(\mu_0, \Sigma_0)$ on the mean μ_k and an inverse-Wishart IW(ν, Δ) prior on the covariance Σ_k.

Conditioned on the state sequence $z_{1:T}$, we may examine the set of observations assigned to state k, which we denote by $Y_k = \{y_t \mid z_t = k\}$. The posterior distributions over the mean and covariance parameters of that state are then derived from the standard inverse-Wishart and Gaussian posterior distributions of Sec. 2.4.3 to be:

$$\begin{aligned} \Sigma_k \mid \mu_k &\sim \mathrm{IW}(\bar{\nu}_k \bar{\Delta}_k, \bar{\nu}_k) \\ \mu_k \mid \Sigma_k &\sim \mathcal{N}(\bar{\mu}_k, \bar{\Sigma}_k), \end{aligned} \tag{B.13}$$

where,

$$\begin{aligned}
\bar{\nu}_k &= \nu + |Y_k| \\
\bar{\nu}_k \bar{\Delta}_k &= \nu\Delta + \sum_{t \in Y_k} (y_t - \mu_k)(y_t - \mu_k)' \\
\bar{\Sigma}_k &= (\Sigma_0^{-1} + |Y_k|\Sigma_k^{-1})^{-1} \\
\bar{\mu}_k &= \bar{\Sigma}_k \left(\Sigma_0^{-1}\mu_0 + \Sigma_k \sum_{t \in Y_k} y_t \right).
\end{aligned}$$

The sampler alternates between sampling μ_k given Σ_k and Σ_k given μ_k several times before moving on to the next stage in the sampling algorithm. The equations for the sticky HDP-HMM with DP emissions follow directly by considering $Y_{k,j} = \{y_t \mid z_t = k, s_t = j\}$ when resampling parameter $\theta_{k,j} = \{\mu_{k,j}, \Sigma_{k,j}\}$.

Appendix C

Hyperparameters

In this appendix we present the derivations of the conditional distributions for the hyperparameters of the sticky HDP-HMM of Chapter 3. These hyperparameters include α, κ, γ, σ, and λ, where λ is considered fixed. Many of these derivations follow directly from those presented in [40, 162].

We parameterize our model by $(\alpha+\kappa)$ and $\rho = \kappa/(\alpha+\kappa)$; this simplifies the resulting sampler. We place Gamma(a, b) priors on each of the concentration parameters $(\alpha+\kappa)$, γ, and σ, and a Beta(c, d) prior on ρ. The a and b parameters of the gamma hyperprior may differ for each of the concentration parameters. In the following sections, we derive the resulting posterior distribution of these hyperparameters.

■ C.1 Posterior of $(\alpha + \kappa)$

Let us assume that there are J restaurants in the franchise at a given iteration of the sampler. Note that for the HDP-HMM, the number of restaurants is equal to the number of unique states in $z_{1:T}$[1]. As depicted in Fig. 2.13(b), the generative model dictates that for each restaurant j we have $\tilde{\pi}_j \sim \text{GEM}(\alpha+\kappa)$, and a table assignment is determined for each customer by $t_{ji} \sim \tilde{\pi}_j$. In total there are $n_{j\cdot}$ draws from this stick-breaking measure over table assignments resulting in $m_{j\cdot}$ unique tables. By Eq. (2.218) and using the fact that the restaurants are mutually conditionally independent, we may write:

$$\begin{aligned}
&p(\alpha+\kappa \mid m_{1\cdot},\dots,m_{J\cdot},n_{1\cdot},\dots,n_{J\cdot}) \propto p(\alpha+\kappa)p(m_{1\cdot},\dots,m_{J\cdot} \mid \alpha+\kappa,n_{1\cdot},\dots,n_{J\cdot})\\
&\quad\propto p(\alpha+\kappa)\prod_{j=1}^{J} p(m_{j\cdot} \mid \alpha+\kappa,n_{j\cdot}) \propto p(\alpha+\kappa)\prod_{j=1}^{J} s(n_{j\cdot},m_{j\cdot})(\alpha+\kappa)^{m_{j\cdot}}\frac{\Gamma(\alpha+\kappa)}{\Gamma(\alpha+\kappa+n_{j\cdot})}\\
&\quad\propto p(\alpha+\kappa)(\alpha+\kappa)^{m_{\cdot\cdot}}\prod_{j=1}^{J}\frac{\Gamma(\alpha+\kappa)}{\Gamma(\alpha+\kappa+n_{j\cdot})}. \qquad \text{(C.1)}
\end{aligned}$$

[1] One must account for the fact that the initial state is drawn from a special initial distribution, and the value of z_T does not create a new restaurant.

Using the fact that the gamma function has the property $\Gamma(z+1) = z\Gamma(z)$ and is related to the beta function via $\beta(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x+y)$, we rewrite this distribution as

$$\begin{aligned}
&p(\alpha+\kappa \mid m_{1\cdot}, \ldots, m_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}) \\
&\qquad \propto p(\alpha+\kappa)(\alpha+\kappa)^{m_{\cdot\cdot}} \prod_{j=1}^{J} \frac{(\alpha+\kappa+n_{j\cdot})\beta(\alpha+\kappa+1, n_{j\cdot})}{(\alpha+\kappa)\Gamma(n_{j\cdot})} \\
&\qquad = p(\alpha+\kappa)(\alpha+\kappa)^{m_{\cdot\cdot}} \prod_{j=1}^{J} \left(1 + \frac{n_{j\cdot}}{\alpha+\kappa}\right) \int_0^1 r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1} dr_j, \qquad \text{(C.2)}
\end{aligned}$$

where the second equality arises from the fact that $\beta(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt$. We introduce a set of auxiliary random variables $r = \{r_1, \ldots, r_J\}$, where each $r_j \in [0, 1]$. Now, we augment the posterior with these auxiliary variables as follows:

$$\begin{aligned}
&p(\alpha+\kappa, r \mid m_{1\cdot}, \ldots, m_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}) \\
&\qquad \propto p(\alpha+\kappa)(\alpha+\kappa)^{m_{\cdot\cdot}} \prod_{j=1}^{J} \left(1 + \frac{n_{j\cdot}}{\alpha+\kappa}\right) r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1} \\
&\qquad \propto (\alpha+\kappa)^{a+m_{\cdot\cdot}-1} e^{-(\alpha+\kappa)b} \prod_{j=1}^{J} \left(1 + \frac{n_{j\cdot}}{\alpha+\kappa}\right) r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1} \\
&\qquad = (\alpha+\kappa)^{a+m_{\cdot\cdot}-1} e^{-(\alpha+\kappa)b} \prod_{j=1}^{J} \sum_{s_j \in \{0,1\}} \left(\frac{n_{j\cdot}}{\alpha+\kappa}\right)^{s_j} r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1}. \qquad \text{(C.3)}
\end{aligned}$$

Here, we have used the fact that we placed a Gamma(a, b) prior on $(\alpha + \kappa)$. We add another set of auxiliary variables $s = \{s_1, \ldots, s_J\}$, with each $s_j \in \{0, 1\}$, to further simplify this distribution. The joint distribution over $(\alpha + \kappa)$, r, and s is given by

$$\begin{aligned}
&p(\alpha+\kappa, r, s \mid m_{1\cdot}, \ldots, m_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}) \\
&\qquad \propto (\alpha+\kappa)^{a+m_{\cdot\cdot}-1} e^{-(\alpha+\kappa)b} \prod_{j=1}^{J} \left(\frac{n_{j\cdot}}{\alpha+\kappa}\right)^{s_j} r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1}. \qquad \text{(C.4)}
\end{aligned}$$

The conditional distribution of $\alpha + \kappa$ given the auxiliary variables is:

$$\begin{aligned}
&p(\alpha+\kappa \mid r, s, m_{1\cdot}, \ldots, m_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}) \\
&\qquad \propto (\alpha+\kappa)^{a+m_{\cdot\cdot}-1-\sum_{j=1}^{J} s_j} e^{-(\alpha+\kappa)(b-\sum_{j=1}^{J} \log r_j)} \\
&\qquad = \text{Gamma}(a + m_{\cdot\cdot} - \sum_{j=1}^{J} s_j, b - \sum_{j=1}^{J} \log r_j), \qquad \text{(C.5)}
\end{aligned}$$

while the auxiliary variables have conditional distributions:

$$\begin{aligned} p(r_j \mid \alpha+\kappa, r_{\setminus j}, s, m_{1\cdot}, \dots, m_{J\cdot}, n_{1\cdot}, \dots, n_{J\cdot}) &\propto r_j^{\alpha+\kappa}(1-r_j)^{n_{j\cdot}-1} \\ &= \mathrm{Beta}(\alpha+\kappa+1, n_{j\cdot}) \qquad \text{(C.6)} \\ p(s_j \mid \alpha+\kappa, r, s_{\setminus j}, m_{1\cdot}, \dots, m_{J\cdot}, n_{1\cdot}, \dots, n_{J\cdot}) &\propto \left(\frac{n_{j\cdot}}{\alpha+\kappa}\right)^{s_j} \\ &= \mathrm{Ber}\left(\frac{n_{j\cdot}}{n_{j\cdot}+\alpha+\kappa}\right). \qquad \text{(C.7)} \end{aligned}$$

■ C.2 Posterior of γ

We may similarly derive the conditional distribution of γ. The generative model depicted in Fig. 2.13(b) dictates that $\beta \sim \mathrm{GEM}(\gamma)$ and that each table t considers ordering a dish $\bar{k}_{jt} \sim \beta$. From Eq. (A.21), we see that the sampled value $\bar{m}_{j\cdot}$ represents the total number of tables in restaurant j where the considered dish $\bar{k}_{jt}$ was the served dish k_{jt} (i.e., the number of tables with considered dishes that were not overridden.) Thus, $\bar{m}_{\cdot\cdot}$ is the total number of *informative* draws from β. If K is the number of unique *served* dishes, which can be inferred from $z_{1:T}$, then the number of unique *considered* dishes at the informative tables is:

$$\bar{K} = \sum_{k=1}^{K} \mathbf{1}(\bar{m}_{\cdot k} > 0) = K - \sum_{k=1}^{K} \mathbf{1}(\bar{m}_{\cdot k} = 0 \text{ and } m_{kk} > 0). \qquad \text{(C.8)}$$

We use the notation $\mathbf{1}(A)$ to represent an indicator random variable that is 1 if the event A occurs and 0 otherwise. The only case where $\bar{K}$ is not equivalent to K is if every instance of a served dish k arose from an override in restaurant k and this dish was never considered in any other restaurant. That is, there were no informative considerations of dish k, implying $\bar{m}_{\cdot k} = 0$, while dish k was served in restaurant k, implying $m_{kk} > 0$ so that k is counted in K. This is equivalent to counting how many dishes k had an informative table consider ordering dish k, regardless of the restaurant. We may now use Eq. (2.218) to form the conditional distribution on γ:

$$\begin{aligned} p(\gamma \mid \bar{K}, \bar{m}_{\cdot\cdot}) &\propto p(\gamma)p(\bar{K} \mid \gamma, \bar{m}_{\cdot\cdot}) \\ &\propto p(\gamma)s(\bar{m}_{\cdot\cdot}, \bar{K})\gamma^{\bar{K}}\frac{\Gamma(\gamma)}{\Gamma(\gamma+\bar{m}_{\cdot\cdot})} \\ &\propto p(\gamma)\gamma^{\bar{K}}\frac{(\gamma+\bar{m}_{\cdot\cdot})\beta(\gamma+1, \bar{m}_{\cdot\cdot})}{\gamma\Gamma(\bar{m}_{\cdot\cdot})} \\ &\propto p(\gamma)\gamma^{\bar{K}-1}(\gamma+\bar{m}_{\cdot\cdot})\int_0^1 \eta^{\gamma}(1-\eta)^{\bar{m}_{\cdot\cdot}-1}d\eta. \qquad \text{(C.9)} \end{aligned}$$

As before, we introduce an auxiliary random variable $\eta \in [0,1]$ so that the joint distribution over γ and η can be written as

$$\begin{aligned} p(\gamma, \eta \mid \bar{K}, \bar{m}_{\cdot\cdot}) &\propto p(\gamma)\gamma^{\bar{K}-1}(\gamma + \bar{m}_{\cdot\cdot})\eta^{\gamma}(1-\eta)^{\bar{m}_{\cdot\cdot}-1} \\ &\propto \gamma^{a+\bar{K}-2}(\gamma + \bar{m}_{\cdot\cdot})e^{-\gamma(b-\log\eta)}(1-\eta)^{\bar{m}_{\cdot\cdot}-1}. \end{aligned} \tag{C.10}$$

Here, we have used the fact that there is a Gamma(a,b) prior on γ. We may add an indicator random variable $\zeta \in \{0,1\}$ as we did in Eq. (C.4), such that

$$p(\gamma, \eta, \zeta \mid \bar{K}, \bar{m}_{\cdot\cdot}) \propto \gamma^{a+\bar{K}-1}\left(\frac{\bar{m}_{\cdot\cdot}}{\gamma}\right)^{\zeta} e^{-\gamma(b-\log\eta)}(1-\eta)^{\bar{m}_{\cdot\cdot}-1}. \tag{C.11}$$

The resulting conditional distributions are given by:

$$\begin{aligned} p(\gamma \mid \eta, \zeta, \bar{K}, \bar{m}_{\cdot\cdot}) &\propto \gamma^{a+\bar{K}-1-\zeta}e^{-\gamma(b-\log\eta)} \\ &= \text{Gamma}(a + \bar{K} - \zeta, b - \log\eta) & \text{(C.12)} \\ p(\eta \mid \gamma, \zeta, \bar{K}, \bar{m}_{\cdot\cdot}) &\propto \eta^{\gamma}(1-\eta)^{\bar{m}_{\cdot\cdot}-1} = \text{Beta}(\gamma+1, \bar{m}_{\cdot\cdot}) & \text{(C.13)} \\ p(\zeta \mid \gamma, \eta, \bar{K}, \bar{m}_{\cdot\cdot}) &\propto \left(\frac{\bar{m}_{\cdot\cdot}}{\gamma}\right)^{\zeta} = \text{Ber}\left(\frac{\bar{m}_{\cdot\cdot}}{\bar{m}_{\cdot\cdot} + \gamma}\right). & \text{(C.14)} \end{aligned}$$

Alternatively, we can directly identify Eq (C.10) as leading to a conditional distribution on γ that is a simple mixture of two Gamma distributions:

$$\begin{aligned} p(\gamma \mid \eta, \bar{K}, \bar{m}_{\cdot\cdot}) &\propto \gamma^{a+\bar{K}-2}(\gamma + \bar{m}_{\cdot\cdot})e^{-\gamma(b-\log\eta)} \\ &\propto \pi_{\bar{m}}\text{Gamma}(a + \bar{K}, b - \log\eta) & \text{(C.15)} \\ &\quad +(1-\pi_{\bar{m}})\text{Gamma}(a + \bar{K} - 1, b - \log\eta) \\ p(\eta \mid \gamma, \bar{K}, \bar{m}_{\cdot\cdot}) &\propto \eta^{\gamma}(1-\eta)^{\bar{m}_{\cdot\cdot}-1} = \text{Beta}(\gamma+1, \bar{m}_{\cdot\cdot}), & \text{(C.16)} \end{aligned}$$

where

$$\pi_{\bar{m}} = \frac{a + \bar{K} - 1}{\bar{m}_{\cdot\cdot}(b - \log\eta)}. \tag{C.17}$$

The distribution in Eq. (C.3) would lead to a much more complicated mixture of gamma distributions. The addition of auxiliary variables s_j greatly simplifies the interpretation of the distribution.

■ C.3 Posterior of σ

The derivation of the conditional distribution on σ is similar to that of $(\alpha + \kappa)$ in that we have J distributions $\psi_j \sim \text{GEM}(\sigma)$. The state-specific mixture component index is generated as $s_t \sim \psi_{z_t}$ implying that we have $n_{j\cdot}$ total draws from ψ_j, one for each occurrence of $z_t = j$. Let K'_j be the number of unique mixture components associated with these draws from ψ_j. Then, after adding auxiliary variables r' and s', the conditional distributions of σ and these auxiliary variables are:

$$\begin{aligned} p(\sigma \mid r', s', K'_{1\cdot}, \dots, K'_{J\cdot}, n_{1\cdot}, \dots, n_{J\cdot}) & \\ & \propto (\sigma)^{a+K'_{\cdot\cdot}-1-\sum_{j=1}^{J} s'_j} e^{-(\sigma)(b-\sum_{j=1}^{J} \log r'_j)} \qquad \text{(C.18)} \\ p(r'_j \mid \sigma, r'_{\setminus j}, s', K'_{1\cdot}, \dots, K'_{J\cdot}, n_{1\cdot}, \dots, n_{J\cdot}) & \propto r_j'^{\sigma}(1-r'_j)^{n_{j\cdot}-1} \qquad \text{(C.19)} \\ p(s'_j \mid \sigma, r', s'_{\setminus j}, K'_{1\cdot}, \dots, K'_{J\cdot}, n_{1\cdot}, \dots, n_{J\cdot}) & \propto \left(\frac{n_{j\cdot}}{\sigma}\right)^{s'_j}. \qquad \text{(C.20)} \end{aligned}$$

In practice, it is useful to alternate between sampling the auxiliary variables and concentration parameters α, γ, and σ for several iterations before moving to sampling the other variables of this model.

■ C.4 Posterior of ρ

Finally, we derive the conditional distribution of ρ. We have $m_{\cdot\cdot} = \sum_k m_{\cdot k}$ total draws of $w_{jt} \sim \text{Ber}(\rho)$, with $\sum_j w_{j\cdot}$ the number of Bernoulli successes. Here, each success represents a table's considered dish being overridden by the house specialty dish. Using these facts, and the Beta(c,d) prior on ρ, we have

$$\begin{aligned} p(\rho \mid \boldsymbol{w}) & \propto p(\boldsymbol{w} \mid \rho) p(\rho) \\ & \propto \begin{pmatrix} m_{\cdot\cdot} \\ \sum_j w_{j\cdot} \end{pmatrix} \rho^{\sum_j w_{j\cdot}} (1-\rho)^{m_{\cdot\cdot}-\sum_j w_{j\cdot}} \frac{\Gamma(c+d)}{\Gamma(c)\Gamma(d)} \rho^{c-1} (1-\rho)^{d-1} \\ & \propto \rho^{\sum_j w_{j\cdot}+c-1} (1-\rho)^{m_{\cdot\cdot}-\sum_j w_{j\cdot}+d-1} \\ & \propto \text{Beta}\left(\sum_j w_{j\cdot} + c, m_{\cdot\cdot} - \sum_j w_{j\cdot} + d \right). \qquad \text{(C.21)} \end{aligned}$$

Appendix D

HDP-SLDS and HDP-AR-HMM Message Passing

In this appendix, we explore the computation of the backwards message passing and forward sampling scheme used for generating samples of the mode sequence $z_{1:T}$ and state sequence $\boldsymbol{x}_{1:T}$ in the HDP-AR-HMM and HDP-SLDS models of Chapter 4.

■ D.1 Mode Sequence Message Passing for Blocked Sampling

Consider a switching VAR(r) process. To derive the forward-backward procedure for jointly sampling the mode sequence $z_{1:T}$ given observations $\boldsymbol{y}_{1:T}$, plus r initial observations $\boldsymbol{y}_{1-r:0}$, we first note that the chain rule and Markov structure allows us to decompose the joint distribution as follows:

$$\begin{aligned} p(z_{1:T} \mid \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(z_T \mid z_{T-1}, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_{T-1} \mid z_{T-2}, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ \cdots p(z_2 \mid z_1, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}). \end{aligned} \tag{D.1}$$

Thus, we may first sample z_1 from $p(z_1 \mid \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta})$, then condition on this value to sample z_2 from $p(z_2 \mid z_1, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta})$, and so on. The conditional distribution of z_1 is derived as:

$$\begin{aligned} p(z_1 \mid \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) &\propto p(z_1) p(\boldsymbol{y}_1 \mid \theta_{z_1}, \boldsymbol{y}_{1-r:0}) \sum_{z_{2:T}} \prod_t p(z_t \mid \pi_{z_{t-1}}) p(\boldsymbol{y}_t \mid \theta_{z_t}, \boldsymbol{y}_{t-r:t-1}) \\ &\propto p(z_1) p(\boldsymbol{y}_1 \mid \theta_{z_1}, \boldsymbol{y}_{1-r:0}) \sum_{z_2} p(z_2 \mid \pi_{z_1}) p(\boldsymbol{y}_2 \mid \theta_{z_2}, \boldsymbol{y}_{2-r:1}) m_{3,2}(z_2) \\ &\propto p(z_1) p(\boldsymbol{y}_1 \mid \theta_{z_1}, \boldsymbol{y}_{1-r:0}) m_{2,1}(z_1), \end{aligned} \tag{D.2}$$

where $m_{t,t-1}(z_{t-1})$ is the backward message passed from z_t to z_{t-1} and is recursively defined by:

$$m_{t,t-1}(z_{t-1}) \propto \begin{cases} \sum_{z_t} p(z_t \mid \pi_{z_{t-1}}) p(\boldsymbol{y}_t \mid \theta_{z_t}, \boldsymbol{y}_{t-r:t-1}) m_{t+1,t}(z_t), & t \leq T; \\ 1, & t = T+1. \end{cases} \tag{D.3}$$

The general conditional distribution of z_t is:

$$p(z_t \mid z_{t-1}, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto p(z_t \mid \pi_{z_{t-1}}) p(\boldsymbol{y}_t \mid \theta_{z_t}, \boldsymbol{y}_{t-r:t-1}) m_{t+1,t}(z_t). \tag{D.4}$$

For the HDP-AR-HMM, these distributions are given by:

$$\begin{aligned} p(z_t = k \mid z_{t-1}, \boldsymbol{y}_{1-r:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) &\propto \pi_{z_{t-1}}(k) \mathcal{N}(\boldsymbol{y}_t; \sum_{i=1}^{r} A_i^{(k)} \boldsymbol{y}_{t-i}, \Sigma^{(k)}) m_{t+1,t}(k) \\ m_{t+1,t}(k) &= \sum_{j=1}^{L} \pi_k(j) \mathcal{N}(\boldsymbol{y}_{t+1}; \sum_{i=1}^{r} A_i^{(j)} \boldsymbol{y}_{t-i}, \Sigma^{(j)}) m_{t+2,t+1}(j) \\ m_{T+1,T}(k) &= 1 \quad k = 1, \dots, L. \end{aligned} \tag{D.5}$$

D.2 State Sequence Message Passing for Blocked Sampling

A similar sampling scheme is used for generating samples of the state sequence $\boldsymbol{x}_{1:T}$. Although we now have a continuous state space, the computation of the backwards messages $m_{t+1,t}(\boldsymbol{x}_t)$ is still analytically feasible since we are working with Gaussian densities. Assume, $m_{t+1,t}(\boldsymbol{x}_t) \propto \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t+1,t}, \Lambda_{t+1,t})$, where $\mathcal{N}^{-1}(x; \theta, \Lambda)$ denotes a Gaussian distribution on x in information form with mean $\mu = \Lambda^{-1}\theta$ and covariance $\Sigma = \Lambda^{-1}$. Given a fixed mode sequence $z_{1:T}$, we simply have a time-varying linear dynamic system. Using similar derivations to those of Sec. 2.7.5, the backwards messages for the HDP-SLDS can be recursively defined by

$$m_{t,t-1}(\boldsymbol{x}_{t-1}) \propto \int_{\mathcal{X}_t} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, z_t) p(\boldsymbol{y}_t | \boldsymbol{x}_t) m_{t+1,t}(\boldsymbol{x}_t) d\boldsymbol{x}_t. \tag{D.6}$$

For this model, the state transition density of Eq. (D.6) can be expressed as

$$\begin{aligned} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, z_t) &\propto \exp\left\{ -\frac{1}{2} (\boldsymbol{x}_t - A^{(z_t)} \boldsymbol{x}_{t-1} - \boldsymbol{\mu}^{(z_t)})^T \Sigma^{-(z_t)} (\boldsymbol{x}_t - A^{(z_t)} \boldsymbol{x}_{t-1} - \boldsymbol{\mu}^{(z_t)}) \right\} \\ &\propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} A^{(z_t)^T} \Sigma^{-(z_t)} A^{(z_t)} & -A^{(z_t)^T} \Sigma^{-(z_t)} \\ -\Sigma^{-(z_t)} A^{(z_t)} & \Sigma^{-(z_t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \right. \\ &\qquad \left. + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} -A^{(z_t)^T} \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} \\ \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} \end{bmatrix} \right\}. \end{aligned} \tag{D.7}$$

We can similarly write the likelihood in exponentiated quadratic form

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{x}_t) &\propto \exp\left\{-\frac{1}{2}(\boldsymbol{y}_t - C\boldsymbol{x}_t)^T R^{-1}(\boldsymbol{y}_t - C\boldsymbol{x}_t)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & C^T R^{-1} C \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \right. \\ &\qquad\qquad \left. + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 \\ C^T R^{-1} \boldsymbol{y}_t \end{bmatrix}\right\}, \end{aligned} \tag{D.8}$$

as well as the messages

$$\begin{aligned} m_{t+1,t}(\boldsymbol{x}_t) &\propto \exp\left\{-\frac{1}{2}\boldsymbol{x}_t^T \Lambda_{t+1,t}\boldsymbol{x}_t + \boldsymbol{x}_t^T \theta_{t+1,t}\right\} \\ &\propto \exp\left\{-\frac{1}{2}\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{t+1,t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 \\ \theta_{t+1,t} \end{bmatrix}\right\}. \end{aligned} \tag{D.9}$$

The product of these quadratics is given by:

$$\begin{aligned} &p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, z_t)p(\boldsymbol{y}_t|\boldsymbol{x}_t)m_{t+1,t}(\boldsymbol{x}_t) \propto \\ &\exp\left\{-\frac{1}{2}\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} A^{(z_t)^T}\Sigma^{-(z_t)}A & -A^{(z_t)^T}\Sigma^{-(z_t)} \\ -\Sigma^{-(z_t)}A^{(z_t)} & \Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \right. \\ &\qquad \left. + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} -A^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} \\ C^T R^{-1}\boldsymbol{y}_t + \Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} + \theta_{t+1,t} \end{bmatrix}\right\} \end{aligned} \tag{D.10}$$

Using standard Gaussian marginalization identities we integrate over $\boldsymbol{x}_t$ to get,

$$m_{t,t-1}(\boldsymbol{x}_{t-1}) \propto \mathcal{N}^{-1}(\boldsymbol{x}_{t-1}; \theta_{t,t-1}, \Lambda_{t,t-1}), \tag{D.11}$$

where,

$$\begin{aligned} \theta_{t,t-1} &= -A^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} + A^{(z_t)^T}\Sigma^{-(z_t)}(\Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t})^{-1} \\ &\qquad\qquad \cdot (C^T R^{-1}\boldsymbol{y}_t + \Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} + \theta_{t+1,t}) \\ \Lambda_{t,t-1} &= A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} - A^{(z_t)^T}\Sigma^{-(z_t)}(\Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t})^{-1}\Sigma^{-(z_t)}A^{(z_t)}. \end{aligned} \tag{D.12}$$

The backwards message passing recursion is initialized with $m_{T+1,T} \sim \mathcal{N}^{-1}(\boldsymbol{x}_T; 0, 0)$. Let,

$$\begin{aligned} \Lambda^b_{t|t} &= C^T R^{-1} C + \Lambda_{t+1,t} \\ \theta^b_{t|t} &= C^T R^{-1}\boldsymbol{y}_t + \theta_{t+1,t}. \end{aligned} \tag{D.13}$$

Then we can define the following recursion, which we note is equivalent to a backwards running Kalman filter in information form,

$$\begin{aligned}
\Lambda^b_{t-1|t-1} &= C^T R^{-1} C + A^{(z_t)^T} \Sigma^{-(z_t)} A^{(z_t)} \\
&\quad - A^{(z_t)^T} \Sigma^{-(z_t)} (\Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t})^{-1} \Sigma^{-(z_t)} A^{(z_t)} \\
&= C^T R^{-1} C + A^{(z_t)^T} \Sigma^{-(z_t)} A^{(z_t)} - A^{(z_t)^T} \Sigma^{-(z_t)} (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1} \Sigma^{-(z_t)} A^{(z_t)} \\
\theta^b_{t-1|t-1} &= C^T R^{-1} \boldsymbol{y}_{t-1} - A^{(z_t)^T} \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} + A^{(z_t)^T} \Sigma^{-(z_t)} (\Sigma^{-(z_t)} + C^T R^{-1} C + \Lambda_{t+1,t})^{-1} \\
&\qquad \cdot (C^T R^{-1} \boldsymbol{y}_t + \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} + \theta_{t+1,t}) \\
&= C^T R^{-1} \boldsymbol{y}_{t-1} - A^{(z_t)^T} \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} \\
&\quad + A^{(z_t)^T} \Sigma^{-(z_t)} (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1} (\theta^b_{t|t} + \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)})
\end{aligned} \tag{D.14}$$

We initialize at time T with

$$\begin{aligned}
\Lambda^b_{T|T} &= C^T R^{-1} C \\
\theta^b_{T|T} &= C^T R^{-1} \boldsymbol{y}_T
\end{aligned} \tag{D.15}$$

An equivalent, but more numerically stable recursion is summarized in Algorithm 19.

After computing the messages $m_{t+1,t}(\boldsymbol{x}_t)$ backwards in time, we sample the state sequence $\boldsymbol{x}_{1:T}$ working forwards in time. As with the discrete mode sequence, one can decompose the posterior distribution of the state sequence as

$$\begin{aligned}
p(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) &= p(\boldsymbol{x}_T \mid \boldsymbol{x}_{T-1}, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) p(\boldsymbol{x}_{T-1} \mid \boldsymbol{x}_{T-2}, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) \\
&\quad \cdots p(\boldsymbol{x}_2 \mid \boldsymbol{x}_1, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) p(\boldsymbol{x}_1 \mid \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}).
\end{aligned} \tag{D.16}$$

where

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) \propto p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, A^{(z_t)}, \Sigma^{(z_t)}, \boldsymbol{\mu}^{(z_t)}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, R) m_{t+1,t}(\boldsymbol{x}_t). \tag{D.17}$$

For the HDP-SLDS, the product of these distributions is equivalent to

$$\begin{aligned}
p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:T}, z_{1:T}, \boldsymbol{\theta}) &\propto \mathcal{N}(\boldsymbol{x}_t; A^{(z_t)} \boldsymbol{x}_{t-1} + \boldsymbol{\mu}^{(z_t)}, \Sigma^{(z_t)}) \mathcal{N}(\boldsymbol{y}_t; C \boldsymbol{x}_t, R) m_{t+1,t}(\boldsymbol{x}_t) \\
&\propto \mathcal{N}(\boldsymbol{x}_t; A^{(z_t)} \boldsymbol{x}_{t-1} + \boldsymbol{\mu}^{(z_t)}, \Sigma^{(z_t)}) \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^b_{t|t}, \Lambda^b_{t|t}) \\
&\propto \mathcal{N}^{-1}(\boldsymbol{x}_t; \Sigma^{-(z_t)}(A^{(z_t)} \boldsymbol{x}_{t-1} + \boldsymbol{\mu}^{(z_t)}) + \theta^b_{t|t}, \Sigma^{-(z_t)} + \Lambda^b_{t|t}),
\end{aligned} \tag{D.18}$$

which is a simple Gaussian distribution so that the normalization constant is easily computed. Specifically, for each $t \in \{1, \dots, T\}$ we sample $\boldsymbol{x}_t$ from

$$\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_t; (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1} (\Sigma^{-(z_t)} A^{(z_t)} \boldsymbol{x}_{t-1} + \Sigma^{-(z_t)} \boldsymbol{\mu}^{(z_t)} + \theta^b_{t|t}), (\Sigma^{-(z_t)} + \Lambda^b_{t|t})^{-1}). \tag{D.19}$$

```
1. Initialize filter with
       $\Lambda^b_{T|T} = C^T R^{-1} C$
       $\theta^b_{T|T} = C^T R^{-1} \boldsymbol{y}_T$
2. Working backwards in time, for each $t \in \{T-1, \dots, 1\}$:
   (a) Compute
       $\tilde{J}_{t+1} = \Lambda^b_{t+1|t+1}(\Lambda^b_{t+1|t+1} + \Sigma^{-(z_{t+1})})^{-1}$
       $\tilde{L}_{t+1} = I - \tilde{J}_{t+1}.$
   (b) Predict
       $\Lambda_{t+1,t} = A^{(z_{t+1})^T}(\tilde{L}_{t+1}\Lambda^b_{t+1|t+1}\tilde{L}^T_{t+1} + \tilde{J}_{t+1}\Sigma^{-(z_{t+1})}\tilde{J}^T_{t+1})A^{(z_{t+1})}$
       $\theta_{t+1,t} = A^{(z_{t+1})^T}\tilde{L}_{t+1}(\theta^b_{t+1|t+1} - \Lambda^b_{t+1|t+1}\boldsymbol{\mu}^{(z_{t+1})})$
   (c) Update
       $\Lambda^b_{t|t} = \Lambda_{t+1,t} + C^T R^{-1} C$
       $\theta^b_{t|t} = \theta_{t+1,t} + C^T R^{-1} \boldsymbol{y}_t$
3. Set
       $\Lambda^b_{0|0} = \Lambda_{1,0}$
       $\theta^b_{0|0} = \theta_{1,0}$
```

Algorithm 19. Numerically stable form of the backwards Kalman information filter.

■ D.3 Mode Sequence Message Passing for Sequential Sampling

A similar sampling scheme to Carter and Kohn [26] is used for generating samples of the mode sequence $z_{1:T}$ having marginalized over the state sequence $\boldsymbol{x}_{1:T}$. Specifically, we sample z_t from:

$$\begin{aligned} p(z_t = k \mid z_{\backslash t}, \boldsymbol{y}_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) &\propto p(z_t = k \mid z_{\backslash t}, \boldsymbol{\pi}) p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\backslash t}) \\ &\propto \pi_{z_{t-1}}(k)\pi_k(z_{t+1}) p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\backslash t}). \end{aligned} \tag{D.20}$$

We omit the dependency on $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ for compactness. To compute the likelihood for each z_t, we combine forward and backward messages along with the local dynamics and measurements as follows:

$$\begin{aligned}p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\setminus t}) &\propto \int_{\mathcal{X}_{t-1}} \int_{\mathcal{X}_t} m_{t-2,t-1}(\boldsymbol{x}_{t-1}) p(\boldsymbol{y}_{t-1} \mid \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, z_t = k) \\ &\qquad\qquad p(\boldsymbol{y}_t|\boldsymbol{x}_t) m_{t+1,t}(\boldsymbol{x}_t) d\boldsymbol{x}_t d\boldsymbol{x}_{t-1} \qquad (\text{D.21}) \\ &\propto \int_{\mathcal{X}_t} \int_{\mathcal{X}_{t-1}} m_{t-2,t-1}(\boldsymbol{x}_{t-1}) p(\boldsymbol{y}_{t-1} \mid \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, z_t = k) d\boldsymbol{x}_{t-1} \\ &\qquad\qquad p(\boldsymbol{y}_t|\boldsymbol{x}_t) m_{t+1,t}(\boldsymbol{x}_t) d\boldsymbol{x}_t, \qquad (\text{D.22})\end{aligned}$$

where the backwards messages are defined as in Appendix D.2 and the forward messages by:

$$m_{t-1,t}(\boldsymbol{x}_t) \propto \int_{\mathcal{X}_{t-1}} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, z_t) p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1}) m_{t-2,t-1}(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1}. \tag{D.23}$$

To derive the forward message passing recursions, assume that

$$m_{t-2,t-1}(\boldsymbol{x}_{t-1}) \propto \mathcal{N}^{-1}(\boldsymbol{x}_{t-1}; \theta_{t-2,t-1}, \Lambda_{t-2,t-1}) \tag{D.24}$$

and z_t is known. The terms of the integrand of Eq. (D.23) can be written as:

$$\begin{aligned}p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, z_t) &= \mathcal{N}(\boldsymbol{x}_t; A^{(z_t)}\boldsymbol{x_{t-1}} + \boldsymbol{\mu}^{(z_t)}, \Sigma^{(z_t)}) \qquad (\text{D.25}) \\ &\propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}^T \begin{bmatrix} \Sigma^{-(z_t)} & -\Sigma^{-(z_t)}A^{(z_t)} \\ -A^{(z_t)^T}\Sigma^{-(z_t)} & A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} \right. \\ &\qquad \left. + \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}^T \begin{bmatrix} \Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} \\ -A^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} \end{bmatrix} \right\} \quad .\end{aligned}$$

$$\begin{aligned}m_{t-2,t-1}(\boldsymbol{x}_{t-1})p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1}) &\propto \mathcal{N}(\boldsymbol{x}_{t-1}; \Lambda^{-f}_{t-1|t-1}\theta^f_{t-1|t-1}, \Lambda^f_{t-1|t-1}) \qquad (\text{D.26}) \\ \propto \exp\left\{ -\frac{1}{2} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & \Lambda^f_{t-1|t-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} \right. &+ \left. \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}^T \begin{bmatrix} 0 \\ \theta^f_{t-1|t-1} \end{bmatrix} \right\},\end{aligned}$$

where, similar to the backwards recursions, we have made the following definitions

$$\begin{aligned}\theta^f_{t|t} &= \theta_{t-1,t} + C^T R^{-1}\boldsymbol{y}_t \\ \Lambda^f_{t|t} &= \Lambda_{t-1,t} + C^T R^{-1} C.\end{aligned} \tag{D.27}$$

Combining these distributions and integrating over $\boldsymbol{x}_{t-1}$, we have

$$m_{t-1,t}(\boldsymbol{x}_t) \propto \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t-1,t}, \Lambda_{t-1,t}) \tag{D.28}$$

with

$$\begin{aligned}\theta_{t-1,t} &= \Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} \\ &\quad + \Sigma^{-(z_t)}A^{(z_t)}(A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} + \Lambda^f_{t-1|t-1})^{-1}(\theta^f_{t-1|t-1} - A^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)}) \\ \Lambda_{t-1,t} &= \Sigma^{-(z_t)} - \Sigma^{-(z_t)}A^{(z_t)}(A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} + \Lambda^f_{t-1|t-1})^{-1}A^{(z_t)^T}\Sigma^{-(z_t)},\end{aligned} \tag{D.29}$$

or equivalently,

$$\begin{aligned}\theta_{t-1,t} &= \Lambda_{t-1,t}(\boldsymbol{\mu}^{(z_t)} + A^{(z_t)}\Lambda^{-f}_{t-1|t-1}\theta^f_{t-1|t-1}) \\ \Lambda_{t-1,t} &= (\Sigma^{(z_t)} + A^{(z_t)}\Lambda^{-f}_{t-1|t-1}A^{(z_t)^T})^{-1}.\end{aligned} \tag{D.30}$$

Assuming $\boldsymbol{x}_0 \sim \mathcal{N}(0, P_0)$, we initialize at time $t = 0$ to

$$\begin{aligned}\theta_{-1,0} &= 0 \\ \Lambda_{-1,0} &= P_0^{-1}.\end{aligned} \tag{D.31}$$

An equivalent, but more numerically stable recursion is summarized in Algorithm 20. However, this algorithm relies on the dynamic matrix $A^{(k)}$ being invertible.

1. Initialize filter with

$$\Lambda^b_{0|0} = P_0$$
$$\theta^b_{0|0} = 0$$

2. Working forwards in time, for each $t \in \{1, \dots, T\}$:

   (a) Compute

$$\begin{aligned}M_t &= A^{-(z_{t+1})^T}\Lambda^{-f}_{t|t}A^{-(z_{t+1})} \\ J_t &= M_t(M_t + \Sigma^{-(z_{t+1})})^{-1} \\ L_t &= I - J_t.\end{aligned}$$

   (b) Predict

$$\begin{aligned}\Lambda_{t-1,t} &= L_{t-1}M_{t-1}L^T_{t-1} + J_{t-1}\Sigma^{-(z_t)}J^T_{t-1} \\ \theta_{t-1,t} &= L_{t-1}A^{-(z_t)^T}(\theta^f_{t-1|t-1} + \theta^f_{t-1|t-1}A^{-(z_t)}\boldsymbol{\mu}^{(z_t)})\end{aligned}$$

   (c) Update

$$\begin{aligned}\Lambda^f_{t|t} &= \Lambda_{t-1,t} + C^TR^{-1}C \\ \theta^f_{t|t} &= \theta_{t-1,t} + C^TR^{-1}\boldsymbol{y}_t\end{aligned}$$

Algorithm 20. Numerically stable form of the forward Kalman information filter.

We now return to the computation of the likelihood of Eq. (D.22). We note that the integral over $\boldsymbol{x}_{t-1}$ is equivalent to computing the message $m_{t-1,t}(\boldsymbol{x}_t)$ using $z_t = k$.

However, we have to be careful that any constants that were previously ignored in this message passing are not a function of z_t. For the meantime, let us assume that there exists such a constant and let us denote this special message by

$$m_{t-1,t}(\boldsymbol{x}_t; z_t) \propto c(z_t)\mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t-1,t}(z_t), \Lambda_{t-1,t}(z_t)). \tag{D.32}$$

Then, the likelihood can be written as

$$p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\setminus t}) \propto \int_{\mathcal{X}_t} m_{t-1,t}(\boldsymbol{x}_t; z_t = k) p(\boldsymbol{y}_t|\boldsymbol{x}_t) m_{t+1,t}(\boldsymbol{x}_t) d\boldsymbol{x}_t \tag{D.33}$$

$$\propto \int_{\mathcal{X}_t} c(k)\mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t-1,t}(k), \Lambda_{t-1,t}(k))\mathcal{N}^{-1}(\boldsymbol{x}_t; \theta^b_{t|t}, \Lambda^b_{t|t}) d\boldsymbol{x}_t \tag{D.34}$$

Combining the information parameters, and maintaining the term in the normalizing constant that is a function of k, this is equivalent to

$$p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\setminus t}) \propto c(k)|\Lambda_{t-1,t}(k)|^{1/2} \exp\left(-\frac{1}{2}\theta_{t-1,t}(k)^T \Lambda_{t-1,t}(k)^{-1}\theta_{t-1,t}(k)\right)$$
$$\int_{\mathcal{X}_t} \exp\left(-\frac{1}{2}\boldsymbol{x}_t^T(\Lambda_{t-1,t}(k) + \Lambda^b_{t|t})\boldsymbol{x}_t + \boldsymbol{x}_t^T(\theta_{t-1,t}(k) + \theta^b_{t|t})\right) d\boldsymbol{x}_t \tag{D.35}$$

To compute this integral, we write the integrand in terms of a Gaussian distribution times a constant. The integral is then simply that constant term:

$$p(\boldsymbol{y}_{1:T} \mid z_t = k, z_{\setminus t}) \propto c(k)|\Lambda_{t-1,t}(k)|^{1/2} \exp\left(-\frac{1}{2}\theta_{t-1,t}(k)^T \Lambda_{t-1,t}(k)^{-1}\theta_{t-1,t}(k)\right)$$
$$|\Lambda_{t-1,t}(k) + \Lambda^b_{t|t}|^{-1/2} \exp\left(\frac{1}{2}(\theta_{t-1,t}(k) + \theta^b_{t|t})^T(\Lambda_{t-1,t}(k) + \Lambda^b_{t|t})^{-1}(\theta_{t-1,t}(k) + \theta^b_{t|t})\right)$$
$$\int_{\mathcal{X}_t} \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t-1,t}(k) + \theta^b_{t|t}, \Lambda_{t-1,t}(k) + \Lambda^b_{t|t}) d\boldsymbol{x}_t$$
$$\propto c(k)\frac{|\Lambda_{t-1,t}(k)|^{1/2}}{|\Lambda_{t-1,t}(k) + \Lambda^b_{t|t}|^{1/2}}$$
$$\exp\bigg(-\frac{1}{2}\theta_{t-1,t}(k)^T \Lambda_{t-1,t}(k)^{-1}\theta_{t-1,t}(k)$$
$$+\frac{1}{2}(\theta_{t-1,t}(k) + \theta^b_{t|t})^T(\Lambda_{t-1,t}(k) + \Lambda^b_{t|t})^{-1}(\theta_{t-1,t}(k) + \theta^b_{t|t})\bigg)$$

Thus,

$$p(z_t = k \mid z_{\setminus t}, \boldsymbol{y}_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto \pi_{z_{t-1}}(k)\pi_k(z_{t+1})c(k)|\Lambda_t^{(k)}|^{1/2}|\Lambda_t^{(k)} + \Lambda^b_{t|t}|^{-1/2}$$
$$\exp\left(-\frac{1}{2}\theta_t^{(k)^T}\Lambda_t^{-(k)}\theta_t^{(k)} + \frac{1}{2}(\theta_t^{(k)} + \theta^b_{t|t})^T(\Lambda_t^{(k)} + \Lambda^b_{t|t})^{-1}(\theta_t^{(k)} + \theta^b_{t|t})\right) \tag{D.36}$$

We now show that $c(z_t)$ is not a function z_t. The only place where the previously ignored dependency on z_t arises is from $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, z_t)$. Namely,

$$\begin{aligned} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, z_t) &= \frac{\exp(-\frac{1}{2}\boldsymbol{\mu}^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)})}{|\Sigma^{(z_t)}|^{1/2}} \cdot \text{exponential}_1 \\ &= c_1(z_t) \cdot \text{exponential}_1 \end{aligned} \tag{D.37}$$

where exponential_1 is the exponentiated quadratic of Eq. (D.25). Then, when compute the message $m_{t-1,t}(\boldsymbol{x}_t; z_t)$ we update the previous message $m_{t-2,t-1}(\boldsymbol{x}_{t-1})$ by incorporating the local likelihood $p(\boldsymbol{y}_{t-1} \mid \boldsymbol{x}_{t-1})$ and then propagating the state estimate with $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, z_t)$ and integrating over $\boldsymbol{x}_{t-1}$. Namely, we combine the distribution of Eq. (D.37) with the exponentiated quadratic of Eq. (D.26) and integrate over $\boldsymbol{x}_{t-1}$:

$$m_{t-1,t}(\boldsymbol{x}_t; z_t) \propto c_1(z_t) \int_{\mathcal{X}_{t-1}} \text{exponential}_1 \cdot \text{exponential}_2 d\boldsymbol{x}_{t-1}, \tag{D.38}$$

where exponential_2 is the exponentiated quadratic of Eq. (D.26).

Since $m_{t-2,t-1}(\boldsymbol{x}_{t-1}) \propto p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-2}, z_{1:t-1})$, and the Markov properties of the state space model dictate

$$\begin{aligned} p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1}, z_{1:t-1}) &= p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-2}, z_{1:t-1}) \\ &\propto p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1})m_{t-2,t-1}(\boldsymbol{x}_{t-1}), \end{aligned} \tag{D.39}$$

then

$$p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1}, z_{1:t-1}) = c_2 \cdot \text{exponential}_2.$$

We note that the normalizing constant c_2 is not a function of z_t since we have only considered z_τ for $\tau < t$.

Once again exploiting the conditional independencies induced by the Markov structure of our state space model, and plugging in Eq. (D.37) and Eq. (D.40),

$$\begin{aligned} p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1}, z_{1:t}) &= p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_{t-1}, z_t)p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1}, z_{1:t-1}) \\ &= (c_1(z_t) \cdot \text{exponential}_1)(c_2 \cdot \text{exponential}_2) \\ &= c_1(z_t)c_2 \cdot \text{exponential}_1 \cdot \text{exponential}_2. \end{aligned} \tag{D.40}$$

Plugging this results into Eq. (D.38), we have

$$\begin{aligned} m_{t-1,t}(\boldsymbol{x}_t; z_t) &\propto c_1(z_t) \int_{\mathcal{X}_{t-1}} \frac{1}{c_1(z_t)c_2} p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1}, z_{1:t}) d\boldsymbol{x}_{t-1} \\ &\propto \frac{1}{c_2} p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1}, z_{1:t}). \end{aligned} \tag{D.41}$$

Comparing Eq. (D.41) to Eq. (D.32), and noting that

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1}, z_{1:t}) = \mathcal{N}^{-1}(\boldsymbol{x}_t; \theta_{t-1,t}(z_t), \Lambda_{t-1,t}(z_t)),$$

we see that $c(z_t) = \frac{1}{c_2}$ and is thus not a function of z_t.

Algebraically, we could derive this result as follows.

$$\begin{aligned} m_{t-1,t}(\boldsymbol{x}_t; z_t) &\propto c_1(z_t) \int_{\mathcal{X}_{t-1}} \text{exponential}_1 \cdot \text{exponential}_2 d\boldsymbol{x}_{t-1} \\ &= c_1(z_t)c_3(z_t) \int_{\mathcal{X}_{t-1}} \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}; \boldsymbol{\Lambda}(z_t)^{-1}\boldsymbol{\theta}(z_t), \boldsymbol{\Lambda}(z_t) \right) d\boldsymbol{x}_{t-1}, \end{aligned} \tag{D.42}$$

where $\boldsymbol{\theta}(z_t)$ and $\boldsymbol{\Lambda}(z_t)$ are the information parameters determined by combining the functional forms of exponential$_1$ and exponential$_2$, and

$$c_1(z_t)c_3(z_t) = \frac{\exp\{-\frac{1}{2}\boldsymbol{\mu}^{(z_t)^T}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)}\}}{|\Sigma^{(z_t)}|^{1/2}} \frac{\exp\{\frac{1}{2}\boldsymbol{\theta}(z_t)^T\boldsymbol{\Lambda}(z_t)^{-1}\boldsymbol{\theta}(z_t)\}}{|\boldsymbol{\Lambda}(z_t)|^{1/2}}. \tag{D.43}$$

Computing these terms in parts, and using standard linear algebra properties of block matrices,

$$\begin{aligned} |\boldsymbol{\Lambda}(z_t)| &= |\Sigma^{-(z_t)}||(A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} + \Lambda^f_{t-1|t-1}) - A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)}| \\ &= |\Sigma^{-(z_t)}||\Lambda^f_{t-1|t-1}| \end{aligned} \tag{D.44}$$

$$\begin{aligned} \boldsymbol{\Lambda}(z_t)^{-1} &= \begin{bmatrix} (\Sigma^{-(z_t)} - \Sigma^{-(z_t)}A^{(z_t)}\tilde{\boldsymbol{\Lambda}}(z_t)^{-1}A^{(z_t)^T}\Sigma^{-(z_t)})^{-1} & A^{(z_t)}\Lambda^f_{t-1|t-1} \\ \Lambda^f_{t-1|t-1}A^{(z_t)^T} & (\tilde{\boldsymbol{\Lambda}}(z_t) - A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)})^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \Sigma^{(z_t)} + A^{(z_t)}\Lambda^{-f}_{t-1|t-1}A^{(z_t)^T} & A^{(z_t)}\Lambda^f_{t-1|t-1} \\ \Lambda^f_{t-1|t-1}A^{(z_t)^T} & \Lambda^{-f}_{t-1|t-1} \end{bmatrix}, \end{aligned} \tag{D.45}$$

where $\tilde{\boldsymbol{\Lambda}}(z_t) = (A^{(z_t)^T}\Sigma^{-(z_t)}A^{(z_t)} + \Lambda^f_{t-1|t-1})$ and we have used the matrix inversion lemma in obtaining the last equality. Using this form of $\boldsymbol{\Lambda}(z_t)^{-1}$, we readily obtain

$$\boldsymbol{\theta}(z_t)^T\boldsymbol{\Lambda}(z_t)^{-1}\boldsymbol{\theta}(z_t) = \boldsymbol{\mu}^{(z_t)}\Sigma^{-(z_t)}\boldsymbol{\mu}^{(z_t)} + \theta^{f^T}_{t-1|t-1}\Lambda^{-f}_{t-1|t-1}\theta^f_{t-1|t-1}. \tag{D.46}$$

Thus,

$$c_1(z_t)c_3(z_t) = \frac{\exp\{\frac{1}{2}\theta^{f^T}_{t-1|t-1}\Lambda^{-f}_{t-1|t-1}\theta^f_{t-1|t-1}\}}{|\Lambda^f_{t-1|t-1}|^{1/2}}, \tag{D.47}$$

which does not depend upon the value of z_t.

Appendix E

Derivation of Maneuvering Target Tracking Sampler

In this Appendix we derive the maneuvering target tracking (MTT) sampler outlined in Sec. 4.3.2. Many of the derivations follow directly from those of Appendice A and D, and we refer to sections of that appendix as appropriate. Recall the MTT model:

$$\begin{aligned} z_t &\sim \pi_{z_{t-1}} \\ \boldsymbol{x}_t &= A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t(z_t) + \boldsymbol{v}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + \boldsymbol{w}_t, \end{aligned} \tag{E.1}$$

where

$$\boldsymbol{u}_t(k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}) \quad \boldsymbol{v}_t \sim \mathcal{N}(0, Q) \quad \boldsymbol{w}_t \sim \mathcal{N}(0, R). \tag{E.2}$$

As described in Sec. 4.3.2, we are interested in jointly sampling the control input and dynamical mode $(\boldsymbol{u}_t, z_t)$, marginalizing over the state sequence $\boldsymbol{x}_{1:T}$, the transition distributions $\boldsymbol{\pi}$, and the dynamic parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}\}$. One can factor the desired conditional distribution factorizes as,

$$p(\boldsymbol{u}_t, z_t | z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \beta, \alpha, \kappa, \lambda) = p(z_t | z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \beta, \alpha, \kappa, \lambda) p(\boldsymbol{u}_t | z_{1:T}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \lambda). \tag{E.3}$$

The distribution in Eq.(E.3) is a hybrid distribution: each discrete value of the dynamical mode indicator variable z_t corresponds to a different continuous distribution on the control input $\boldsymbol{u}_t$. We analyze each of the conditional distributions of Eq. (E.3) by considering the joint distribution on all of the model parameters, and then marginalizing $\boldsymbol{x}_{1:T}$, $\boldsymbol{\pi}$, and θ_k. (Note that marginalization over θ_j for $j \neq k$ simply results in a constant.)

$$\begin{aligned} p(z_t = k | z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \beta, \alpha, \kappa, \lambda) &\propto \int_{\boldsymbol{\pi}} \prod_j p(\pi_j | \beta, \alpha, \kappa) \prod_\tau p(z_\tau | \pi_{z_{\tau-1}}) d\pi \\ \int_{\mathcal{U}_t} \int p(\theta_k \mid \lambda) \prod_{\tau | z_\tau = k} p(\boldsymbol{u}_\tau | \theta_k) d\theta_k &\int_{\mathcal{X}} \prod_\tau p(\boldsymbol{x}_\tau | \boldsymbol{x}_{\tau-1}, \boldsymbol{u}_\tau) p(\boldsymbol{y}_\tau | \boldsymbol{x}_\tau) d\boldsymbol{x}_{1:T} d\boldsymbol{u}_t. \end{aligned} \tag{E.4}$$

Similarly, we can write the conditional density of $\boldsymbol{u}_t$ for each candidate z_t as,

$$\begin{aligned} p(\boldsymbol{u}_t | z_t = k, z_{\setminus t}, \boldsymbol{u}_{\setminus t}, \boldsymbol{y}_{1:T}, \lambda) \propto \int p(\theta_k \mid \lambda) & \prod_{\tau | z_\tau = k} p(\boldsymbol{u}_\tau | \theta_k) d\theta_k \\ & \int_{\mathcal{X}} \prod_{\tau} p(\boldsymbol{x}_\tau | \boldsymbol{x}_{\tau-1}, \boldsymbol{u}_\tau) p(\boldsymbol{y}_\tau | \boldsymbol{x}_\tau) d\boldsymbol{x}_{1:T}. \end{aligned} \tag{E.5}$$

In the following sections, we evaluate each of these integrals in turn.

■ E.1 Chinese Restaurant Franchise

The integration over $\boldsymbol{\pi}$ appearing in the first line of Eq. (E.4) results in exactly the same predictive distribution as Eq. (A.10) of the sticky HDP-HMM. Namely,

$$\begin{aligned} & p(z_t = k \mid z_{\setminus t}, \beta, \alpha, \kappa) \\ & \quad \propto \begin{cases} (\alpha\beta_k + n_{z_{t-1}k}^{-t} + \kappa\delta(z_{t-1}, k)) \\ \qquad \left(\frac{\alpha\beta_{z_{t+1}} + n_{kz_{t+1}}^{-t} + \kappa\delta(k, z_{t+1}) + \delta(z_{t-1}, k)\delta(k, z_{t+1})}{\alpha + n_{k\cdot}^{-t} + \kappa + \delta(z_{t-1}, k)} \right) & k \in \{1, \dots, K\} \\ \frac{\alpha^2 \beta_{\tilde{k}} \beta_{z_{t+1}}}{\alpha + \kappa} & k = K+1. \end{cases} \end{aligned} \tag{E.6}$$

■ E.2 Normal-Inverse-Wishart Posterior Update

The marginalization of θ_k, appearing both in Eq. (E.4) and Eq. (E.5), can be rewritten as follows:

$$\begin{aligned} \int p(\theta_k | \lambda) \prod_{\tau | z_\tau = k} p(\boldsymbol{u}_\tau | \theta_k) d\theta_k &= \int p(\boldsymbol{u}_t | \theta_k) p(\theta_k | \lambda) \prod_{\tau | z_\tau = k, \tau \neq t} p(\boldsymbol{u}_\tau | \theta_k) d\theta_k \\ &\propto \int p(\boldsymbol{u}_t | \theta_k) p(\theta_k | \{\boldsymbol{u}_\tau | z_\tau = k, \tau \neq t\}, \lambda) d\theta_k \\ &= p(\boldsymbol{u}_t | \{\boldsymbol{u}_\tau | z_\tau = k, \tau \neq t\}, \lambda). \end{aligned} \tag{E.7}$$

Here, the set $\{\boldsymbol{u}_\tau | z_\tau = k, \tau \neq t\}$ denotes all the observations $\boldsymbol{u}_\tau$ other than $\boldsymbol{u}_t$ that were drawn from the Gaussian parameterized by θ_k. When θ_k has a normal-inverse-Wishart prior $\mathcal{NIW}(\kappa, \boldsymbol{\vartheta}, \nu, \Delta)$, we use the results of Sec. 2.4.3 to derive that

$$p(\boldsymbol{u}_t | \{\boldsymbol{u}_\tau | z_\tau = k, \tau \neq t\}, \kappa, \boldsymbol{\vartheta}, \nu, \Delta) \simeq \mathcal{N}\left(\boldsymbol{u}_t; \bar{\boldsymbol{\vartheta}}, \frac{(\bar{\kappa} + 1)\bar{\nu}}{\bar{\kappa}(\bar{\nu} - d - 1)} \bar{\Delta} \right) \triangleq \mathcal{N}(\boldsymbol{u}_t; \hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k), \tag{E.8}$$

where

$$\begin{aligned}
\bar{\kappa} &= \kappa + |\{\boldsymbol{u}_s | z_s = k, s \neq t\}| \\
\bar{\nu} &= \nu + |\{\boldsymbol{u}_s | z_s = k, s \neq t\}| \\
\bar{\kappa}\bar{\boldsymbol{\vartheta}} &= \kappa\boldsymbol{\vartheta} + \sum_{\boldsymbol{u}_s \in \{\boldsymbol{u}_s | z_s = k, s \neq t\}} \boldsymbol{u}_s \\
\bar{\nu}\bar{\Delta} &= \nu\Delta + \sum_{\boldsymbol{u}_s \in \{\boldsymbol{u}_s | z_s = k, s \neq t\}} \boldsymbol{u}_s\boldsymbol{u}_s^T + \kappa\boldsymbol{\vartheta}\boldsymbol{\vartheta}^T - \bar{\kappa}\bar{\boldsymbol{\vartheta}}\bar{\boldsymbol{\vartheta}}^T
\end{aligned} \tag{E.9}$$

Here, we are using the moment-matched Gaussian approximation to the Student-t predictive distribution for $\boldsymbol{u}_t$ induced by marginalizing θ_k.

E.3 Marginalization by Message Passing

When considering the control input $\boldsymbol{u}_t$ and conditioning on the values of all $\boldsymbol{u}_\tau$, $\tau \neq t$, the marginalization over all states $\boldsymbol{x}_{1:T}$ can be equated to a message passing scheme that relies on the conditionally linear dynamical system induced by fixing $\boldsymbol{u}_\tau$, $\tau \neq t$. Specifically,

$$\begin{aligned}
&\int_{\mathcal{X}} \prod_\tau p(\boldsymbol{x}_\tau | \boldsymbol{x}_{\tau-1}, \boldsymbol{u}_\tau) p(\boldsymbol{y}_\tau | \boldsymbol{x}_\tau) dx \\
&\quad \propto \int_{\mathcal{X}_{t-1}} \int_{\mathcal{X}_t} m_{t-1,t-2}(\boldsymbol{x}_{t-1}) p(\boldsymbol{y}_{t-1} | \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{u}_t) p(\boldsymbol{y}_t | \boldsymbol{x}_t) m_{t,t+1}(\boldsymbol{x}_t) dx_t dx_{t-1} \\
&\quad \propto p(\boldsymbol{y}_{1:T} | \boldsymbol{u}_t; \boldsymbol{u}_{\setminus t}),
\end{aligned} \tag{E.10}$$

where we recall the definitions of the forward messages $m_{t-1,t}(\boldsymbol{x}_t)$ and backward messages $m_{t+1,t}(\boldsymbol{x}_t)$ from Appendix D.2. For our MTT model of Eq. (E.1), however, instead of accounting for a process noise mean $\boldsymbol{\mu}^{(z_\tau)}$ at time τ in the filtering equations, we must account for the control input $\boldsymbol{u}_\tau$. Conditioning on $\boldsymbol{u}_\tau$, one can equate $B\boldsymbol{u}_\tau$ with a process noise mean, and thus we simply replace $\boldsymbol{\mu}^{(z_\tau)}$ with $B\boldsymbol{u}_\tau$ in the filtering equations of Appendix D.2. Similarly, we replace the process noise covariance term $\Sigma^{(z_\tau)}$ with our process noise covariance Q. (Note that although $\boldsymbol{u}_\tau(z_\tau) \sim \mathcal{N}(\boldsymbol{\mu}^{(z_\tau)}, \Sigma^{(z_\tau)})$, we condition on the value $\boldsymbol{u}_\tau$ so that the MTT parameters $\{\boldsymbol{\mu}^{(z_\tau)}, \Sigma^{(z_\tau)}\}$ do not factor into the message passing equations.)

E.4 Combining Messages

To compute the likelihood of Eq. (E.10), we take the filtered estimates of $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$, combine them with the local dynamics and local likelihood, and marginalize over $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$. To aid in this computation, we consider the exponentiated quadratic form of each term in the integrand of Eq. (E.10). We then join these terms and use standard Gaussian integration formulas to arrive at the desired likelihood. The derivation of this likelihood greatly parallels that for the sequential mode sequence sampler of Appendix D.3.

Recall the forward filter recursions of Appendix D.2 in terms of information parameters

$$\{\theta_{t-1,t}, \Lambda_{t-1,t}, \theta^f_{t|t}, \Lambda^f_{t|t}\},$$

and the backward filter recursions in terms of

$$\{\theta_{t+1,t}, \Lambda_{t+1,t}, \theta^b_{t|t}, \Lambda^b_{t|t}\}.$$

Replace $\boldsymbol{\mu}^{(z_t)}$ with $B\boldsymbol{u}_t$ and $\Sigma^{(z_t)}$ with Q where appropriate. We many then write $m_{t,t+1}(\boldsymbol{x}_t)$ updated with the likelihood $p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1})$ in exponentiated quadratic form as:

$$\begin{aligned} m_{t-1,t-2}(\boldsymbol{x}_{t-1})p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1}) \\ \propto \exp\Bigg\{ -\frac{1}{2}\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} C^TR^{-1}C + \Lambda_{t-1,t-2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \\ + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} C^TR^{-1}\boldsymbol{y}_{t-1} + \theta_{t-1,t-2} \\ 0 \end{bmatrix} \Bigg\}. \end{aligned}$$

The local dynamics can similarly be written as

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \propto \exp\Bigg\{ -\frac{1}{2}\begin{bmatrix} \boldsymbol{u}_t \\ \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} B^TQ^{-1}B & B^TQ^{-1}A & -B^TQ^{-1} \\ A^TQ^{-1}B & A^TQ^{-1}A & -A^TQ^{-1} \\ -Q^{-1}B & -Q^{-1}A & Q^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_t \\ \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \\ + \begin{bmatrix} \boldsymbol{u}_t \\ \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Bigg\}. \end{aligned}$$

Finally, the backward message $m_{t,t+1}(\boldsymbol{x}_t)$ updated with the likelihood $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ can be written as

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{x}_t)m_{t,t+1}(\boldsymbol{x}_t) \propto \exp\Bigg\{ -\frac{1}{2}\begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & C^TR^{-1}C + \Lambda_{t,t+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix} \\ + \begin{bmatrix} \boldsymbol{x}_{t-1} \\ \boldsymbol{x}_t \end{bmatrix}^T \begin{bmatrix} 0 \\ C^TR^{-1}\boldsymbol{y}_t + \theta_{t,t+1} \end{bmatrix} \Bigg\}. \end{aligned}$$

Using the definitions

$$\begin{aligned} \Lambda^b_{t|t} &= C^TR^{-1}C + \Lambda_{t+1,t} \\ \theta^b_{t|t} &= C^TR^{-1}\boldsymbol{y}_t + \theta_{t+1,t} \\ \Lambda^f_{t|t} &= C^TR^{-1}C + \Lambda_{t-1,t} \\ \theta^f_{t|t} &= C^TR^{-1}\boldsymbol{y}_t + \theta_{t-1,t}, \end{aligned}$$

we may express the entire integrand as

$$m_{t-1,t-2}(\boldsymbol{x}_{t-1})p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{u}_t)p(\boldsymbol{y}_t|\boldsymbol{x}_t)m_{t,t+1}(\boldsymbol{x}_t) \propto$$

$$\exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\\ \boldsymbol{x}_t\end{bmatrix}^T\begin{bmatrix}B^TQ^{-1}B & B^TQ^{-1}A & -B^TQ^{-1}\\ A^TQ^{-1}B & A^TQ^{-1}A+\Lambda^f_{t-1|t-1} & -A^TQ^{-1}\\ -Q^{-1}B & -Q^{-1}A & Q^{-1}+\Lambda^b_{t|t}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\\ \boldsymbol{x}_t\end{bmatrix}\right.$$
$$\left.+\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\\ \boldsymbol{x}_t\end{bmatrix}^T\begin{bmatrix}0\\ \theta^f_{t-1|t-1}\\ \theta^b_{t|t}\end{bmatrix}\right\}$$

Integrating over $\boldsymbol{x}_t$, we obtain an expression proportional to

$$\mathcal{N}^{-1}\left(\begin{bmatrix}\boldsymbol{u}_t^T\\ \boldsymbol{x}_{t-1}\end{bmatrix};\theta\left(\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\end{bmatrix}\right),\Lambda\left(\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\end{bmatrix}\right)\right),$$

with

$$\begin{aligned}\Lambda\left(\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\end{bmatrix}\right) &= \begin{bmatrix}B^TQ^{-1}B & B^TQ^{-1}A\\ A^TQ^{-1}B & A^TQ^{-1}A+\Lambda^f_{t-1|t-1}\end{bmatrix}\\ &\qquad -\begin{bmatrix}B^TQ^{-1}\\ A^TQ^{-1}\end{bmatrix}(Q^{-1}+\Lambda^b_{t|t})^{-1}\begin{bmatrix}Q^{-1}B & Q^{-1}A\end{bmatrix}\\ &= \begin{bmatrix}B^T\Sigma_t^{-1}B & B^T\Sigma_t^{-1}A\\ A^T\Sigma_t^{-1}B & A^T\Sigma_t^{-1}A\end{bmatrix}\\ \theta\left(\begin{bmatrix}\boldsymbol{u}_t\\ \boldsymbol{x}_{t-1}\end{bmatrix}\right) &= \begin{bmatrix}0\\ \theta^f_{t-1|t-1}\end{bmatrix}+\begin{bmatrix}B^TQ^{-1}\\ A^TQ^{-1}\end{bmatrix}(Q^{-1}+\Lambda^b_{t|t})^{-1}\theta^b_{t|t}\\ &= \begin{bmatrix}B^TQ^{-1}K_t^{-1}\theta^b_{t|t}\\ \theta^f_{t-1|t-1}+A^TQ^{-1}K_t^{-1}\theta^b_{t|t}\end{bmatrix}.\end{aligned}$$

Here, we have defined

$$\Sigma_t = Q^{-1}+Q^{-1}(Q^{-1}+\Lambda^b_{t|t})^{-1}Q^{-1} = Q^{-1}+Q^{-1}K_t^{-1}Q^{-1}.$$

Finally, integrating over $\boldsymbol{x}_{t-1}$ yields an expression proportional to

$$\mathcal{N}^{-1}(\boldsymbol{u}_t^T;\theta(\boldsymbol{u}_t),\Lambda(\boldsymbol{u}_t)),$$

with

$$\begin{aligned}\Lambda(\boldsymbol{u}_t) &= B^T\Sigma_t^{-1}B - B^T\Sigma_t^{-1}A(A^T\Sigma_t^{-1}A+\Lambda^f_{t-1|t-1})^{-1}A^T\Sigma_t^{-1}B\\ \theta(\boldsymbol{u}_t) &= B^TQ^{-1}K_t^{-1}\theta^b_{t|t}\\ &\quad - B^T\Sigma_t^{-1}A(A^T\Sigma_t^{-1}A+\Lambda^f_{t-1|t-1})^{-1}(\theta^f_{t-1|t-1}+A^TQ^{-1}K_t^{-1}\theta^b_{t|t}).\end{aligned}$$

■ E.5 Joining Distributions that Depend on $\boldsymbol{u}_t$

We have derived two terms which depend on $\boldsymbol{u}_t$: a prior and a likelihood. Normally, one would consider $p(\boldsymbol{u}_t|\theta_k)$ the prior on $\boldsymbol{u}_t$. However, through marginalization of this parameter, we induced dependencies between the control inputs $\boldsymbol{u}_\tau$ and all the $\boldsymbol{u}_\tau$ that were drawn from a distribution parameterized by θ_k inform us of the distribution over $\boldsymbol{u}_t$. Therefore, we treat $p(\boldsymbol{u}_t|\{\boldsymbol{u}_\tau|z_\tau = k, \tau \neq t\})$ as a prior distribution on $\boldsymbol{u}_t$. The likelihood function $p(\boldsymbol{y}_{1:T}|\boldsymbol{u}_t; \boldsymbol{u}_{\backslash t})$ describes the likelihood of an observation sequence $\boldsymbol{y}_{1:T}$ given the input sequence $\boldsymbol{u}_{1:T}$, containing the random variable is $\boldsymbol{u}_t$.

We multiply the prior distribution by the likelihood function to get the following quadratic expression:

$$\begin{aligned}
&p(\boldsymbol{u}_t|\{\boldsymbol{u}_\tau|z_\tau = k, \tau \neq t\})p(\boldsymbol{y}_{1:T}|\boldsymbol{u}_t; \boldsymbol{u}_{\backslash t}) \\
&\quad\propto \frac{1}{(2\pi)^{N/2}|\hat{\Sigma}_k|^{1/2}} \exp\Bigg\{ -\frac{1}{2}(\boldsymbol{u}_t - \hat{\boldsymbol{\mu}}_k)^T \hat{\Sigma}_k^{-1}(\boldsymbol{u}_t - \hat{\boldsymbol{\mu}}_k) \\
&\qquad\qquad -\frac{1}{2}(\boldsymbol{u}_t - \Lambda(\boldsymbol{u}_t)^{-1}\theta(\boldsymbol{u}_t))^T \Lambda(\boldsymbol{u}_t)(\boldsymbol{u}_t - \Lambda_t y^{-1}\theta(\boldsymbol{u}_t))\Bigg\} \\
&\quad= \frac{1}{(2\pi)^{N/2}|\hat{\Sigma}_k|^{1/2}} \exp\Bigg\{ -\frac{1}{2}\Bigg[\boldsymbol{u}_t^T(\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))\boldsymbol{u}_t - 2\boldsymbol{u}_t^T(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k \\
&\qquad\qquad + \theta(\boldsymbol{u}_t)) + \hat{\boldsymbol{\mu}}_k^T \hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)^T \Lambda(\boldsymbol{u}_t)^{-1}\theta(\boldsymbol{u}_t)\Bigg]\Bigg\} \\
&\quad= \frac{(2\pi)^{N/2}|(\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}|^{1/2}}{(2\pi)^{N/2}|\hat{\Sigma}_k|^{1/2}} \exp\Bigg\{ -\frac{1}{2}\Bigg[\hat{\boldsymbol{\mu}}_k^T \hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)^T \Lambda(\boldsymbol{u}_t)^{-1}\theta(\boldsymbol{u}_t) \\
&\qquad\qquad - (\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t))^T(\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t))\Bigg]\Bigg\} \\
&\qquad \cdot \mathcal{N}(\boldsymbol{u}_t; (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)), (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}) \\
&\quad\triangleq C_k \cdot \mathcal{N}(\boldsymbol{u}_t; (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)), (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}), \qquad \text{(E.11)}
\end{aligned}$$

where we note that the defined constant C_k is a function of $z_t = k$, but not of $\boldsymbol{u}_t$.

■ E.6 Resulting $(\boldsymbol{u}_t, z_t)$ Sampling Distributions

We write Eq. (E.4) and Eq. (E.5) in terms of the derived distributions:

$$\begin{aligned}
p(z_t = k|z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \beta, \alpha, \kappa, \lambda) &\propto p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) \\
&\qquad \int_{\mathcal{U}_t} p(\boldsymbol{u}_t|\{\boldsymbol{u}_\tau|z_\tau = k, \tau \neq t\})p(\boldsymbol{y}_{1:T}|\boldsymbol{u}_t; \boldsymbol{u}_{\backslash t})d\boldsymbol{u}_t, \qquad \text{(E.12)}
\end{aligned}$$

$$p(\boldsymbol{u}_t|z_t = k, z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \lambda) \propto p(\boldsymbol{u}_t|\{\boldsymbol{u}_\tau|z_\tau = k, \tau \neq t\})p(\boldsymbol{y}_{1:T}|\boldsymbol{u}_t; \boldsymbol{u}_{\backslash t}). \qquad \text{(E.13)}$$

Thus, the distribution over z_t, marginalizing $\boldsymbol{u}_t$, is given by

$$\begin{aligned}
p(z_t = k|z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \beta, \alpha, \kappa, \lambda) \\
&\propto p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa) \\
&\qquad \int_{\mathcal{U}_t} C_k \cdot \mathcal{N}(\boldsymbol{u}_t; (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)), (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}) d\boldsymbol{u}_t \\
&\propto C_k \cdot p(z_t = k \mid z_{\backslash t}, \beta, \alpha, \kappa). \qquad \text{(E.14)}
\end{aligned}$$

and the distribution over $\boldsymbol{u}_t$ (for $z_t = k$ fixed) is

$$\begin{aligned}
p(\boldsymbol{u}_t|z_t = k, z_{\backslash t}, \boldsymbol{u}_{\backslash t}, \boldsymbol{y}_{1:T}, \lambda) \\
&= \mathcal{N}(\boldsymbol{u}_t; (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}(\hat{\Sigma}_k^{-1}\hat{\boldsymbol{\mu}}_k + \theta(\boldsymbol{u}_t)), (\hat{\Sigma}_k^{-1} + \Lambda(\boldsymbol{u}_t))^{-1}). \qquad \text{(E.15)}
\end{aligned}$$

Appendix F

Dynamic Parameter Posteriors

In this appendix, we derive the posterior distribution over the dynamic parameters of a switching VAR(r) process defined as follows:

$$\boldsymbol{y}_t = \sum_{i=1}^{r} A_i^{(z_t)} \boldsymbol{y}_{t-i} + \boldsymbol{e}_t(z_t) \quad \boldsymbol{e}_t(k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)}), \tag{F.1}$$

where z_t indexes the mode-specific VAR(r) process at time t. Assume that the state sequence $\{z_1, \dots, z_T\}$ is known and we wish to compute the posterior distribution of the k^{th} mode's VAR(r) parameters $A_i^{(k)}$ for $i = 1, \dots, r$ and $\Sigma^{(k)}$. Let $\{t_1, \dots, t_{N_k}\} = \{t | z_t = k\}$. Then, we may write

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{y}_{t_1} & \boldsymbol{y}_{t_2} & \cdots & \boldsymbol{y}_{t_{N_k}} \end{bmatrix} \\
&= \begin{bmatrix} A_1^{(k)} & A_2^{(k)} & \dots & A_r^{(k)} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_{t_1-1} & \boldsymbol{y}_{t_2-1} & \cdots & \boldsymbol{y}_{t_{N_k}-1} \\ \boldsymbol{y}_{t_1-2} & \boldsymbol{y}_{t_2-2} & \cdots & \boldsymbol{y}_{t_{N_k}-2} \\ \vdots & & & \\ \boldsymbol{y}_{t_1-r} & \boldsymbol{y}_{t_2-r} & \cdots & \boldsymbol{y}_{t_{N_k}-r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_{t_1} & \boldsymbol{e}_{t_2} & \cdots & \boldsymbol{e}_{t_{N_k}} \end{bmatrix}.
\end{aligned} \tag{F.2}$$

We define the following notation for Eq. (F.2):

$$\mathbf{Y}^{(k)} = \mathbf{A}^{(k)} \bar{\mathbf{Y}}^{(k)} + \mathbf{E}^{(k)}, \tag{F.3}$$

and let $\mathbf{D}^{(k)} = \{\mathbf{Y}^{(k)}, \bar{\mathbf{Y}}^{(k)}\}$. In the following sections, we consider two possible priors on the dynamic parameter. In Appendix F.1, we assume that $\boldsymbol{\mu}^{(k)}$ is 0 for all k and consider the conjugate matrix-normal inverse-Wishart (MNIW) prior for $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$. In Appendix F.2, we consider the more general form of Eq. (F.1) and take independent priors on $\mathbf{A}^{(k)}$, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$.

■ F.1 Conjugate Prior — MNIW

To show conjugacy, we place a MNIW prior on the dynamic parameters $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$ and show that the posterior remains MNIW given a set of data from the model of

Eq. (F.1) (assuming $\boldsymbol{\mu}^{(k)} = 0$). The MNIW prior is given by placing a matrix-normal prior $\mathcal{MN}\left(\mathbf{A}^{(k)}; M, \Sigma^{(k)}, K\right)$ on $\mathbf{A}^{(k)}$ given $\Sigma^{(k)}$ (see Eq. (2.94)):

$$p(\mathbf{A}^{(k)} \mid \Sigma^{(k)}) = \frac{|K|^{d/2}}{|2\pi\Sigma^{(k)}|^{m/2}} \exp\left(-\frac{1}{2}tr((\boldsymbol{A} - M)^T\Sigma^{-(k)}(\boldsymbol{A} - M)K)\right) \tag{F.4}$$

and an inverse-Wishart prior $\mathrm{IW}(n_0, S_0)$ on $\Sigma^{(k)}$ (see Eq. (2.95)):

$$p(\Sigma^{(k)}) = \frac{|S_0|^{n_0/2}|\Sigma^{(k)}|^{-(d+n_0+1)/2}}{2^{n_0 d/2}\Gamma_d(n_0/2)} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}S_0)\right) \tag{F.5}$$

where $\Gamma_d(\cdot)$ is the multivariate gamma function and $\boldsymbol{B}^{-(k)}$ denotes $(\boldsymbol{B}^{(k)})^{-1}$ for some matrix $\boldsymbol{B}$.

We first analyze the likelihood of the data, $\mathbf{D}^{(k)}$, given the k^{th} mode's dynamic parameters, $\{\mathbf{A}^{(k)}, \Sigma^{(k)}\}$. Starting with the fact that each observation vector, $\boldsymbol{y}_t$, is conditionally Gaussian given the lag observations, $\bar{\boldsymbol{y}}_t = [\boldsymbol{y}_{t-1}^T \cdots \boldsymbol{y}_{t-r}^T]^T$, we have

$$\begin{aligned}
p(\mathbf{D}^{(k)}|\mathbf{A}^{(k)}, \Sigma^{(k)}) &= \frac{1}{|2\pi\Sigma^{(k)}|^{N_k/2}} \exp\left(-\frac{1}{2}\sum_i(\boldsymbol{y}_{t_i} - \mathbf{A}^{(k)}\bar{\boldsymbol{y}}_{t_i})^T\Sigma^{-(k)}(\boldsymbol{y}_{t_i} - \mathbf{A}^{(k)}\bar{\boldsymbol{y}}_{t_i})\right) \\
&= \frac{1}{|2\pi\Sigma^{(k)}|^{N_k/2}} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})^T)\right) \\
&= \frac{1}{|2\pi\Sigma^{(k)}|^{N_k/2}} \exp\left(-\frac{1}{2}tr((\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})^T\Sigma^{-(k)}(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})\boldsymbol{I})\right) \\
&= \mathcal{MN}\left(\mathbf{Y}^{(k)}; \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)}, \Sigma^{(k)}, \boldsymbol{I}\right).
\end{aligned} \tag{F.6}$$

To derive the posterior of the dynamic parameters, it is useful to first compute

$$p(\mathbf{D}^{(k)}, \mathbf{A}^{(k)} \mid \Sigma^{(k)}) = p(\mathbf{D}^{(k)} \mid \mathbf{A}^{(k)}, \Sigma^{(k)})p(\mathbf{A}^{(k)} \mid \Sigma^{(k)}). \tag{F.7}$$

Using the fact that both the likelihood $p(\mathbf{D}^{(k)} \mid \mathbf{A}^{(k)}, \Sigma^{(k)})$ and the prior $p(\mathbf{A}^{(k)} \mid \Sigma^{(k)})$ are matrix-normally distributed sharing a common parameter $\Sigma^{(k)}$, we have

$$\begin{aligned}
&\log p(\mathbf{D}^{(k)}, \mathbf{A}^{(k)} \mid \Sigma^{(k)}) + C \\
&\quad = -\frac{1}{2}tr((\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})^T\Sigma^{-(k)}(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)}) \\
&\qquad\qquad + (\mathbf{A}^{(k)} - M)^T\Sigma^{-(k)}(\mathbf{A}^{(k)} - M)K) \\
&\quad = -\frac{1}{2}tr(\Sigma^{-(k)}\{(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})(\mathbf{Y}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{Y}}^{(k)})^T \\
&\qquad\qquad + (\mathbf{A}^{(k)} - M)K(\mathbf{A}^{(k)} - M)^T\}) \\
&\quad = -\frac{1}{2}tr(\Sigma^{-(k)}\{\mathbf{A}^{(k)}\mathbf{S}_{\bar{y}\bar{y}}^{(k)}\mathbf{A}^{(k)^T} - 2\mathbf{S}_{y\bar{y}}^{(k)}\mathbf{A}^{(k)^T} + \mathbf{S}_{yy}^{(k)}\}) \\
&\quad = -\frac{1}{2}tr(\Sigma^{-(k)}\{(\mathbf{A}^{(k)} - \mathbf{S}_{y\bar{y}}^{(k)}\mathbf{S}_{\bar{y}\bar{y}}^{-(k)})\mathbf{S}_{\bar{y}\bar{y}}^{(k)}(\mathbf{A}^{(k)} - \mathbf{S}_{y\bar{y}}^{(k)}\mathbf{S}_{\bar{y}\bar{y}}^{-(k)})^T + \mathbf{S}_{y|\bar{y}}^{(k)}\}),
\end{aligned} \tag{F.8}$$

where we have used the definitions:

$$C = -\log \frac{1}{|2\pi\boldsymbol{\Sigma}^{(k)}|^{N_k/2}} \frac{|K|^{d/2}}{|2\pi\boldsymbol{\Sigma}^{(k)}|^{rN_k/2}} \qquad \mathbf{S}^{(k)}_{y|\bar{y}} = \mathbf{S}^{(k)}_{yy} - \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}}\mathbf{S}^{(k)^T}_{y\bar{y}},$$

$$\mathbf{S}^{(k)}_{\bar{y}\bar{y}} = \bar{\mathbf{Y}}^{(k)}\bar{\mathbf{Y}}^{(k)^T} + K \qquad \mathbf{S}^{(k)}_{y\bar{y}} = \mathbf{Y}^{(k)}\bar{\mathbf{Y}}^{(k)^T} + MK \qquad \mathbf{S}^{(k)}_{yy} = \mathbf{Y}^{(k)}\mathbf{Y}^{(k)^T} + MKM^T.$$

Conditioning on the noise covariance $\Sigma^{(k)}$, we see that the dynamic matrix posterior is given by:

$$\begin{aligned}
p(\mathbf{A}^{(k)} \mid \mathbf{D}^{(k)}, \Sigma^{(k)}) &\propto \exp\left(-\frac{1}{2}tr((\mathbf{A}^{(k)} - \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}})^T\Sigma^{-(k)}(\mathbf{A}^{(k)} - \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}})\mathbf{S}^{(k)}_{\bar{y}\bar{y}})\right) \\
&= \mathcal{MN}\left(\mathbf{A}^{(k)}; \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}}, \Sigma^{-(k)}, \mathbf{S}^{(k)}_{\bar{y}\bar{y}}\right). \qquad \text{(F.9)}
\end{aligned}$$

Marginalizing Eq. (F.8) over the dynamic matrix $\mathbf{A}^{(k)}$, we derive

$$\begin{aligned}
p(\mathbf{D}^{(k)} \mid \Sigma^{(k)}) &= \int_{\mathbf{A}^{(k)}} p(\mathbf{D}^{(k)}, \mathbf{A}^{(k)} \mid \Sigma^{(k)}) d\mathbf{A}^{(k)} \\
&= \int_{\mathbf{A}^{(k)}} \frac{|K^{d/2}|}{|2\pi\Sigma^{(k)}|^{N_k/2}|2\pi\Sigma^{(k)}|^{rN_k/2}} \\
&\qquad \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}(\mathbf{A}^{(k)} - \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}})\mathbf{S}^{(k)}_{\bar{y}\bar{y}}(\mathbf{A}^{(k)} - \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}})^T)\right) \\
&\qquad \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}\mathbf{S}^{(k)}_{y|\bar{y}})\right) d\mathbf{A}^{(k)} \\
&= \frac{|K|^{d/2}}{|2\pi\Sigma^{(k)}|^{N_k/2}} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}\mathbf{S}^{(k)}_{y|\bar{y}})\right) \\
&\qquad \int_{\mathbf{A}^{(k)}} \frac{1}{|\mathbf{S}^{(k)}_{\bar{y}\bar{y}}|^{d/2}} \mathcal{MN}\left(\mathbf{A}^{(k)}; \mathbf{S}^{(k)}_{y\bar{y}}\mathbf{S}^{-(k)}_{\bar{y}\bar{y}}, \Sigma^{-(k)}, \mathbf{S}^{(k)}_{\bar{y}\bar{y}}\right) d\mathbf{A}^{(k)} \\
&= \frac{|K|^{d/2}}{|2\pi\Sigma^{(k)}|^{N_k/2}|\mathbf{S}^{(k)}_{\bar{y}\bar{y}}|^{d/2}} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}\mathbf{S}^{(k)}_{y|\bar{y}})\right), \qquad \text{(F.10)}
\end{aligned}$$

which leads us to our final result of the covariance having an inverse-Wishart marginal posterior distribution:

$$\begin{aligned}
p(\Sigma^{(k)} \mid \mathbf{D}^{(k)}) &\propto p(\mathbf{D}^{(k)} \mid \Sigma^{(k)}) p(\Sigma^{(k)}) \\
&\propto \frac{|K|^{d/2}}{|2\pi\Sigma^{(k)}|^{N_k/2}|\mathbf{S}^{(k)}_{\bar{y}\bar{y}}|^{d/2}} \\
&\qquad \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}\mathbf{S}^{(k)}_{y|\bar{y}})\right) |\Sigma^{(k)}|^{-(d+n_0+1)/2} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}S_0)\right) \\
&\propto |\Sigma^{(k)}y|^{-(d+N_k+n_0+1)/2} \exp\left(-\frac{1}{2}tr(\Sigma^{-(k)}(\mathbf{S}^{(k)}_{y|\bar{y}} + S_0))\right) \\
&= \mathrm{IW}(N_k + n_0, \mathbf{S}^{(k)}_{y|\bar{y}} + S_0). \qquad \text{(F.11)}
\end{aligned}$$

F.2 Non-Conjugate Independent Priors on $\mathbf{A}^{(k)}$, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$

In this section, we provide the derivations for the posterior distributions of $\mathbf{A}^{(k)}$, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$ when each of these parameters is given an independent prior.

F.2.1 Normal Prior on $\mathbf{A}^{(k)}$

Assume we place a Gaussian prior, $\mathcal{N}(\boldsymbol{\mu}_A, \Sigma_A)$, on the vectorization of the matrix $\mathbf{A}^{(k)}$, which we denote by $\text{vec}(\mathbf{A}^{(k)})$. To examine the posterior distribution, we first aim to write the data as a linear function of $\text{vec}(\mathbf{A}^{(k)})$. We may rewrite Eq. (F.1) as

$$\begin{aligned} \boldsymbol{y}_t &= \mathbf{A}^{(k)} \begin{bmatrix} \boldsymbol{y}_{t-1} \\ \boldsymbol{y}_{t-2} \\ \vdots \\ \boldsymbol{y}_{t-r} \end{bmatrix} + \boldsymbol{e}_t \qquad \forall t | z_t = k \\ &\triangleq \mathbf{A}^{(k)} \bar{\boldsymbol{y}}_t + \boldsymbol{e}_t(k). \end{aligned} \tag{F.12}$$

Recalling that r is the autoregressive order and d the dimension of the observation vector $\boldsymbol{y}_t$, we can equivalently represent the above as

$$\begin{aligned} &\boldsymbol{y}_t = \boldsymbol{e}_t(k) \\ &+ \begin{bmatrix} \bar{y}_{t,1} & \bar{y}_{t,2} & \cdots & \bar{y}_{t,d*r} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \bar{y}_{t,1} & \bar{y}_{t,2} & \cdots & \bar{y}_{t,d*r} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \bar{y}_{t,1} & \bar{y}_{t,2} & \cdots & \bar{y}_{t,d*r} \end{bmatrix} \begin{bmatrix} a_{1,1}^{(k)} \\ a_{1,2}^{(k)} \\ \vdots \\ a_{1,d*r}^{(k)} \\ a_{2,1}^{(k)} \\ a_{2,2}^{(k)} \\ \vdots \\ a_{d,d*r}^{(k)} \end{bmatrix} \\ &= \begin{bmatrix} \bar{y}_{t,1} I_d & \bar{y}_{t,2} I_d & \cdots & \bar{y}_{t,d*r} I_d \end{bmatrix} \text{vec}(\mathbf{A}^{(k)}) + \boldsymbol{e}_t(k) \triangleq \bar{\boldsymbol{Y}}_t \text{vec}(A) + \boldsymbol{e}_t(k). \end{aligned} \tag{F.13}$$

Here, the columns of $\bar{\boldsymbol{y}}_t$ are permutations of those of the matrix in the first line such that we may write $\boldsymbol{y}_t$ as a function of $\text{vec}(\mathbf{A}^{(k)})$. Noting that $\boldsymbol{e}_t(k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \Sigma^{(k)})$,

$$\begin{aligned} &\log p(\mathbf{D}^{(k)}, \mathbf{A}^{(k)} \mid \Sigma^{(k)}, \boldsymbol{\mu}^{(k)}) \\ &\quad = C - \frac{1}{2} \sum_{t|z_t=k} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)} - \bar{\boldsymbol{Y}}_t \text{vec}(\mathbf{A}^{(k)}))^T \Sigma^{-(k)} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)} - \bar{\boldsymbol{Y}}_t \text{vec}(\mathbf{A}^{(k)})) \\ &\qquad - \frac{1}{2} (\text{vec}(\mathbf{A}^{(k)}) - \boldsymbol{m}_A)^T \Sigma_A^{-1} (\text{vec}(\mathbf{A}^{(k)}) - \boldsymbol{m}_A), \end{aligned} \tag{F.14}$$

which can be rewritten as,

$$\begin{aligned}\log p(\mathbf{D}^{(k)}, \mathbf{A}^{(k)} \mid \Sigma^{(k)}, \boldsymbol{\mu}^{(k)}) = C &- \frac{1}{2}\mathrm{vec}(\mathbf{A}^{(k)})^T \left(\Sigma_A^{-1} + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} \bar{\boldsymbol{Y}}_t\right) \mathrm{vec}(\mathbf{A}^{(k)}) \\ &+ \mathrm{vec}(\mathbf{A}^{(k)})^T \left(\Sigma_A^{-1} \boldsymbol{m}_A + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)})\right) \\ &- \frac{1}{2}\boldsymbol{m}_A^T \Sigma_A^{-1} \boldsymbol{m}_A - \frac{1}{2} \sum_{t|z_t=k} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)})^T \Sigma^{-(k)} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)})\end{aligned} \tag{F.15}$$

Conditioning on the data, we arrive at the desired posterior distribution

$$\begin{aligned}\log p(\mathbf{A}^{(k)} \mid \mathbf{D}^{(k)}, \Sigma^{(k)}, \boldsymbol{\mu}^{(k)}) &= C - \frac{1}{2}\Bigg(\mathrm{vec}(\mathbf{A}^{(k)})^T (\Sigma_A^{-1} + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} \bar{\boldsymbol{Y}}_t) \mathrm{vec}(\mathbf{A}^{(k)}) \\ &\qquad - 2\mathrm{vec}(\mathbf{A}^{(k)})^T (\Sigma_A^{-1} \boldsymbol{m}_A + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)}))\Bigg) \\ &= \mathcal{N}^{-1}\left(\Sigma_A^{-1} \boldsymbol{m}_A + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} (\boldsymbol{y}_t - \boldsymbol{\mu}^{(k)}), \Sigma_A^{-1} + \sum_{t|z_t=k} \bar{\boldsymbol{Y}}_t^T \Sigma^{-(k)} \bar{\boldsymbol{Y}}_t\right)\end{aligned} \tag{F.16}$$

F.2.2 Inverse Wishart Prior on $\Sigma^{(k)}$

We place an inverse-Wishart prior, $\mathrm{IW}(n_0, S_0)$, on $\Sigma^{(k)}$. Let $N_k = |\{t | z_t = k, t = 1, 2, \dots, T\}|$. Conditioned on $\mathbf{A}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$, the standard conjugate prior results presented in Sec. 2.4.3 imply that the posterior of $\Sigma^{(k)}$ is:

$$\begin{aligned}&p(\Sigma^{(k)} \mid \mathbf{D}^{(k)}, \mathbf{A}^{(k)}, \boldsymbol{\mu}^{(k)}) = \\ &\qquad \mathrm{IW}\left(N_k + n_0, S + \sum_{t|z_t=k} (\boldsymbol{y}_t - \mathbf{A}^{(k)}\bar{\boldsymbol{y}}_t - \boldsymbol{\mu}^{(k)})(\boldsymbol{y}_t - \mathbf{A}^{(k)}\bar{\boldsymbol{y}}_t - \boldsymbol{\mu}^{(k)})^T\right).\end{aligned} \tag{F.17}$$

F.2.3 Normal Prior on $\boldsymbol{\mu}^{(k)}$

Finally, we place a Gaussian prior, $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$, on $\boldsymbol{\mu}^{(k)}$. Conditioned on $\mathbf{A}^{(k)}$ and $\Sigma^{(k)}$, the results of Sec. 2.4.3 provide that the posterior of $\boldsymbol{\mu}^{(k)}$ is:

$$\begin{aligned}&p(\boldsymbol{\mu}^{(k)} \mid \mathbf{D}^{(k)}, \mathbf{A}^{(k)}, \Sigma^{(k)}) = \\ &\qquad \mathcal{N}^{-1}\left(\boldsymbol{\mu}^{(k)}; \Sigma_0^{-1}\boldsymbol{\mu}_0 + \Sigma^{-(k)} \sum_{t|z_t=k} (\boldsymbol{y}_t - \mathbf{A}^{(k)}\bar{\boldsymbol{y}}_t), \Sigma_0^{-1} + N_k \Sigma^{-(k)}\right).\end{aligned} \tag{F.18}$$

We iterate between sampling $\mathbf{A}^{(k)}$, $\Sigma^{(k)}$, and $\boldsymbol{\mu}^{(k)}$ many times before moving on to the next step of the Gibbs sampler.

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, chapter 24.1.3, page 824. Models of Neural Networks, III. Dover, 9 edition, 1972.

[2] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[4] B.D.O. Anderson. The realization problem for hidden Markov models. *Mathematics of Control, Signals, and Systems*, 12:80–120, 1999.

[5] C.E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[6] M. Aoki and A. Havenner. State space modeling of multiple time series. *Econometric Reviews*, 10(1):1–59, 1991.

[7] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J.K. Hodgins, and N.S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proc. Graphics Interface*, pages 185–194, 2004.

[8] O. Barndorff-Nielsen. *Information and Exponential Families.* John Wiley, 1978.

[9] M.J. Beal. *Variational Algorithms for Approximate Bayesian Inference.* Ph.D. thesis, University College London, London, UK, 2003.

[10] M.J. Beal and P. Krishnamurthy. Gene expression time course clustering with countably infinite hidden Markov models. In *Proc. Conference on Uncertainty in Artificial Intelligence*, 2006.

[11] M.J. Beal, Z. Ghahramani, and C.E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 14, pages 577–584, 2002.

[12] J.O. Berger and J.M. Bernardo. On the development of reference priors. In J.M. Bernardo, J.O. Berger, D.V. Lindley, and A.F.M. Smith, editors, *Bayesian Statistics 4*, pages 35–60. Oxford University Press, 1992.

[13] J.M. Bernardo. Reference analysis. In D.K. Dey and C.R. Rao, editors, *Handbook of Statistics 25*, pages 17–90. Elsevier, 2005.

[14] J.M. Bernardo. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society B*, 41:113–147, 1979.

[15] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory.* Wiley, 2000.

[16] D.P. Bertsekas and J.N. Tsitsikilis. *Introduction to Probability.* Kluwer Academic, 1996.

[17] R. Bhar and S. Hamori. *Hidden Markov models: applications to financial economics.* Kluwer Academic Publishers, 2004.

[18] D. Blackwell and J.B. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

[19] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

[20] H. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.

[21] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues.* Springer-Verlag, New York, 1999.

[22] P. Buhlmann and J.W. Abraham. Variable length Markov chains. *The Annals of Statistics*, 27(2):480–513, 1999.

[23] F. Caron and A. Doucet. Sparse Bayesian nonparametric regression. In *Proc. International Conference on Machine Learning*, July 2008.

[24] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe. Bayesian inference for dynamic models with Dirichlet process mixtures. In *Proc. International Conference on Information Fusion*, July 2006.

[25] C.K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

[26] C.K. Carter and R. Kohn. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83:589–601, 3 1996.

[27] C.M. Carvalho and H.F. Lopes. Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51:4526–4542, 9 2007.

[28] C.M. Carvalho, J.E. Lucas, Q. Wang, J. Chang, J.R. Nevins, and M. West. High-dimensional sparse factor modelling - Applications in gene expression genomics. *Journal of the American Statistical Association*, 103, 2008.

[29] G. Casella and C. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[30] O.L.V. Costa, M.V. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems.* Springer, 2005.

[31] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 2006.

[32] G. Darmois. Sur les lois de probabilite a estimation exhaustive. *C. R. Acad. Sci. Paris*, 260:1265–1266, 1935.

[33] B. de Finetti. *Funzione Caratteristica Di un Fenomeno Aleatorio*, pages 251–299. 6. Memorie, Classe di Scienze Fisiche, Mathematiche e Naturali. Academia Nazionale dei Lincei, 1931.

[34] P. de Jonga and J. Penzerb. The ARMA model in state space form. *Statistics & Probability Letters*, 70:119–125, 2004.

[35] P. Diaconis and D. Freedman. On the consistency of Bayes estimates. *The Annals of Statistics*, pages 1–26, 1986.

[36] J. Diebolt and C.P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society. Series B*, pages 363–375, 1994.

[37] D. Dunson. Nonparametric Bayes local partition models for random effects. *Biometrika*, 96(2):249–262, 2009.

[38] D. Dunson. Multivariate kernel partition process mixtures. *To appear in Statistica Sinica.*

[39] D. Dunson and J.-H. Park. Kernel stick-breaking processes. *Biometrika*, 95(2): 307–323, 2008.

[40] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[41] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[42] G.D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[43] E.B. Fox, E.B. Sudderth, and A.S. Willsky. Hierarchical Dirichlet processes for tracking maneuvering targets. In *Proc. International Conference on Information Fusion*, July 2007.

[44] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. Nonparametric Bayesian learning of switching dynamical systems. In *Advances in Neural Information Processing Systems*, volume 21, pages 457–464, 2009.

[45] A. Frigessi, P. Di Stefano, C.R. Hwang, and S.J. Sheu. Convergence rates of the Gibbs sampler, the Metropolis algorithm and other single-site updating dynamics. *Journal of the Royal Statistical Society, Series B*, pages 205–219, 1993.

[46] R.G. Gallager. *Discrete Stochastic Processes.* Athena Scientific, 2002.

[47] R. Garcia and P. Perron. An analysis of real interest rate under regime shifts. *Review of Economics & Statistics*, 78(1):111–125, 1996.

[48] J. Gasthaus, F. Wood, D. Gorur, and Y.W. Teh. Dependent Dirichlet process spike sorting. In *Advances in Neural Information Processing Systems*, volume 21, pages 497–504, 2009.

[49] A.E. Gelfand and S. Sahu. Gibbs sampling, identifiability and improper priors in generalized linear mixed models. *Journal American Statistical Association*, 94: 247–253, 1999.

[50] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, pages 457–472, 1992.

[51] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 2004.

[52] Z. Ghahramani and G.E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):831–864, 2000.

[53] S. Ghosal and A. van der Vaart. Posterior convergence rates of Dirichlet mixtures at smooth densities. *Annals of statistics*, 35(2):697, 2007.

[54] S. Ghosal, J.K. Ghosh, and R.V. Ramamoorthi. Posterior consistency of Dirichlet mixtures in density estimation. *Annals of Statistics*, pages 143–158, 1999.

[55] J.K. Ghosh and R.V. Ramamoorthi. *Bayesian nonparametrics.* Springer-Verlag, 2003.

[56] W.R. Gilks and C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society*, 63(1):127–146, 2002.

[57] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Chapman & Hall, 1996.

[58] S. Goldwater, T. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18, pages 459–466, 2006.

[59] D. Görür, F. Jäkel, and C.E. Rasmussen. A choice model with infinitely many latent features. In *Proc. International Conference on Machine learning*, June 2006.

[60] P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[61] J.E. Griffin and M.F.J. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101:179–194, 2006.

[62] T.L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. *Gatsby Computational Neuroscience Unit, Technical Report #2005-001*, 2005.

[63] J.D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

[64] A.C. Harvey, E. Ruiz, and N. Shephard. Multivariate stochastic variance models. *Review of Economic Studies*, 61:247–264, 1994.

[65] D. Heath and W. Sudderth. De Finetti's theorem on exchangeable variables. *The American Statistician*, 30(4):188–189, 1976.

[66] E. Hewitt and L.J. Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, 80(2):470–501, 1955.

[67] N.L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, pages 1259–1294, 1990.

[68] M. Hoffman, P. Cook, and D. Blei. Data-driven recomposition using the hierarchical Dirichlet process hidden Markov model. In *Proc. International Computer Music Conference*, 2008.

[69] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. In *SIGGRAPH*, pages 1082–1089, 2005.

[70] K. Huang, A. Wagner, and Y. Ma. Identification of hybrid linear time-invariant systems via subspace embedding and segmentation SES. In *Proc. IEEE Conference on Decision and Control*, December 2004.

[71] A.T. Ihler, J.W. Fisher III, and A.S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6: 905–936, 2005.

[72] H. Ishwaran and L. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[73] H. Ishwaran and J.S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[74] H. Ishwaran and M. Zarepour. Markov chain Monte Carlo in approximate Dirichlet and beta two–parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

[75] H. Ishwaran and M. Zarepour. Dirichlet prior sieves in finite normal mixtures. *Statistica Sinica*, 12:941–963, 2002.

[76] H. Ishwaran and M. Zarepour. Exact and approximate sum–representations for the Dirichlet process. *Canadian Journal of Statistics*, 30:269–283, 2002.

[77] S. Jain and R.M. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004.

[78] A. Jasra, C.C. Holmes, and D.A. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.

[79] E.T. Jaynes. Prior probabilities. *IEEE Transactions on Systems, Science and Cybernetics*, 4:227–291, 1968.

[80] H. Jeffreys. *Theory of probability.* Oxford University Press, 1998.

[81] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 453–461, 1946.

[82] F. Jelinek. *Statistical methods for speech recognition.* MIT Press, 1998.

[83] M. Johnson. Why doesn't EM find good HMM POS-taggers. In *Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

[84] I.T. Jolliffe. *Principal component analysis.* Springer, 2002.

[85] M.I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155, 2004.

[86] M.I. Jordan. Dirichlet processes, Chinese restaurant processes and all that. In *Tutorial presentation at the NIPS Conference*, 2005.

[87] V.M. Joshi. On the attainment of the Cramér-Rao lower bound. *The Annals of Statistics*, pages 998–1002, 1976.

[88] B.H. Juang and L.R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, pages 251–272, 1991.

[89] T. Kailath, A.H. Sayed, and B. Hassibi, editors. *Linear Estimation.* Prentice Hall, 2000.

[90] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[91] R.E. Kalman. Lyapunov functions for the problem of Lure in automatic control. *Proceedings of the National Academy of Sciences of the United States of America*, 49(2):201–205, 1963.

[92] C. Karlof and D. Wagner. Hidden Markov model cryptanalysis. *Lecture Notes in Computer Science*, pages 17–34, 2003.

[93] A.Y. Khinchine. Korrelationstheorie der stationären stochastiscen prozesse. *Mathematische Annalen*, 109:604–615, 1934.

[94] C.-J. Kim. Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60:1–22, 1994.

[95] J.F.C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

[96] J.F.C. Kingman. *Poisson processes.* Oxford University Press, 1993.

[97] J.J. Kivinen, E.B. Sudderth, and M.I. Jordan. Learning multiscale representations of natural scenes using Dirichlet processes. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[98] B. Koopman. On distributions admitting a sufficient statistic. *Transactions of the American Mathematical Society*, 39:399–409, 1936.

[99] G. Kotsalis, A. Megretski, and M.A. Dahleh. Model reduction of discrete-time Markov jump linear systems. In *Proc. American Control Conference*, June 2006.

[100] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology. applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–1531, 1994.

[101] A. Krogh, B.È. Larsson, G. Von Heijne, and E.L.L. Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580, 2001.

[102] K. Kurihara, M. Welling, and Y.W. Teh. Collapsed variational Dirichlet process mixture models. In *Proc. International Joint Conferences on Artificial Intelligence*, 2007.

[103] S.L. Lauritzen. *Graphical models.* Oxford University Press, USA, 1996.

[104] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[105] N. Lawrence. MATLAB motion capture toolbox. *http://www.cs.man.ac.uk/ neill/mocap/.*

[106] P. Lévy, editor. *Théorie de l'addition des variables aléatoires.* Gauthiers-Villars, 1937.

[107] A. Lindquist and G. Picci. Geometric methods for state space identification. In S. Bittanti and G. Picci, editors, *Identification, Adaptation, Learning*, pages 1–69. Springer, 1994.

[108] J.S. Liu. Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83(3):681–682, 1996.

[109] J.S. Liu, W.H. Wong, and A. Kong. Covariance structure and convergence rate of the Gibbs sampler with various scans. *Journal of the Royal Statistical Society, Series B*, pages 157–169, 1995.

[110] H. Lütkepohl, editor. *New Introduction to Multiple Time Series Analysis.* Springer, 2005.

[111] S.N. MacEachern. Dependent nonparametric processes. In *Proc. Bayesian Statistical Science Section*, pages 50–55, 1998.

[112] D.J.C. MacKay. *Bayesian methods for backprop networks*, chapter 6, pages 211–254. Models of Neural Networks, III. Springer, 1994.

[113] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *The Journal of Machine Learning Research*, 7:2031–2064, 2006.

[114] B. Marthi, H. Pasula, S. Russell, and Y. Peres. Decayed MCMC filtering. In *Proc. Uncertainty in Artificial Intelligence*, August 2002.

[115] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

[116] L.R. Mead and N. Papanicolaou. Maximum entropy in the problem of moments. *Journal of Mathematical Physics*, 25:2404, 1984.

[117] E. Meeds, Z. Ghahramani, R.M. Neal, and S.T. Roweis. Modeling dyadic data with binary latent factors. *Advances in Neural Information Processing Systems*, 19:977–984, 2007.

[118] D. Mochihashi and E. Sumita. The infinite Markov model. In *Advances in Neural Information Processing Systems*, volume 20, pages 1017–1024, 2008.

[119] R.L. Moose, H.F. VanLandingham, and D.H. McCabe. Modeling and estimation of tracking maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 15(3):448–456, 1979.

[120] P. Müller and F.A. Quintana. Nonparametric Bayesian data analysis. *Statistical Science*, 19(1):95–110, 2004.

[121] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[122] K.P. Murphy. Hidden semi-Markov models (HSMMs). *Informal Notes, http://www.cs.ubc.ca/ murphyk/Papers/segment.pdf*, 2002.

[123] K.P. Murphy. Hidden Markov model (HMM) toolbox for MATLAB. *http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html.*

[124] R.M. Neal, editor. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, 1996.

[125] R.M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4):353–366, 1996.

[126] NIST. Rich transcriptions database. *http://www.nist.gov/speech/tests/rt/*, 2007.

[127] F.J. Och and H. Ney. A comparison of alignment models for statistical machine translation. In *Proc. Conference on Computational Linguistics*, volume 2, pages 1086–1090, 2000.

[128] F.J. Och and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.

[129] S. Oh, J. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1–3):103–124, 2008.

[130] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: A tutorial. *European Journal of Control*, 2–3:242–260, 2007.

[131] O. Papaspiliopoulos and G.O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95:169–186, 2008.

[132] V. Pavlović, J.M. Rehg, T.J. Cham, and K.P. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proc. International Conference on Computer Vision*, September 1999.

[133] V. Pavlović, J.M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems*, volume 13, 2001.

[134] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan-Kaufmann, San Mateo, CA, 1988.

[135] M. Petreczky and R. Vidal. Realization theory of stochastic jump-Markov linear systems. In *Proc. IEEE Conference on Decision and Control*, December 2007.

[136] E.J.G. Pitman. Sufficient statistics and intrinsic accuracy. *Proceedings of the Cambridge Philosophical Society*, 32:567–579, 1936.

[137] J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900, 1997.

[138] Z. Psaradakis and N. Spagnolo. Joint determination of the state dimension and autoregressive order for models with Markov regime switching. *Journal of Time Series Analysis*, 27:753–766, 2006.

[139] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[140] L. Ren, A. Patrick, A. Efros, J. Hodgins, and J. Rehg. A data-driven approach to quantifying natural human motion. In *SIGGRAPH*, August 2005.

[141] C.P. Robert. *The Bayesian choice.* Springer, 2007.

[142] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer, 2005.

[143] A. Rodriguez, D.B. Dunson, and A.E. Gelfand. The nested Dirichlet process. *Institute of Statistics and Decision Sciences, Duke University, Technical Report #06-19.*, July 2006.

[144] X. Rong Li and V. Jilkov. Survey of maneuvering target tracking. Part I: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.

[145] X. Rong Li and V. Jilkov. Survey of maneuvering target tracking. Part V: Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[146] C. Ryll-Nardzewski. On stationary sequences of random variables and the de Finettis equivalence. *Colloq. Math.*, 4:149–156, 1957.

[147] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

[148] S.L. Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457): 337–351, 2002.

[149] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4: 639–650, 1994.

[150] R.D. Shachter. Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proc. Conference on Uncertainty in Artificial Intelligence*, pages 480–487, 1998.

[151] G.R. Shafer and P.P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2(1):327–351, 1990.

[152] N. Shephard. Statistical aspects of ARCH and stochastic volatility. In D.R. Cox, D.V. Hinkley, and O.E. Barndorff-Nielsen, editors, *Time Series Models in Econometrics, Finance and Other Fields*, pages 1–67. Chapman & Hall, 1996.

[153] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 4(6): 473–483, 1970.

[154] M.K.P So, K. Lam, and W.K. Li. A stochastic volatility model with Markov switching. *Journal of Business & Economic Statistics*, 16(2):244–253, 1998.

[155] E.L.L. Sonnhammer, G. von Heijne, and A. Krogh. A hidden Markov model for predicting transmembrane helices in protein sequences. In *Proc. International Conference on Intelligent Systems for Molecular Biology*, volume 6, pages 175–82, 1998.

[156] N. Srebro and S. Roweis. Time-varying topic models using dependent Dirichlet processes. *UTML, TR #2005-003*, March 2005.

[157] E.B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking.* Ph.D. thesis, MIT, Cambridge, MA, 2006.

[158] E.B. Sudderth, A. Torralba, W.T. Freeman, and A.S. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77:244–253, 2008.

[159] G.W. Taylor, G.E. Hinton, and S.T. Roweis. Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*, 19: 1345–1352, 2007.

[160] Y.W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[161] Y.W. Teh and M.I. Jordan. Hierarchical bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *To appear in Bayesian Nonparametrics in Practice.* Cambridge University Press.

[162] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[163] Y.W. Teh, D. Gorur, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[164] Y.W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. In *Advances in Neural Information Processing Systems*, volume 20, pages 1481–1488, 2008.

[165] R. Thibaux and M.I. Jordan. Hierarchical beta processes and the Indian buffet process. In *Proc. International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[166] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[167] L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[168] S. Tokdar. Posterior consistency of Dirichlet location-scale mixture of normals in density estimation and regression. *Sankhya: The Indian Journal of Statistics*, 67: 90–110, 2006.

[169] Carnegie Melon University. Graphics lab motion capture database. *http://mocap.cs.cmu.edu/.*

[170] J. Van Gael, Y. Saatci, Y.W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proc. International Conference on Machine Learning*, July 2008.

[171] J. Van Gael, Y.W. Teh, and Z. Ghahramani. The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 21, pages 1697–1704, 2009.

[172] R. Vidal, A. Chiuso, , and S. Soatto. Observability and identifiability of jump linear systems. In *Proc. IEEE Conference on Decision and Control*, December 2002.

[173] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA): Subspace clustering by polynomial factorization, differentiation, and division. *UC Berkeley, Technical Report UCB/ERL.*, August 2003.

[174] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proc. IEEE Conference on Decision and Control*, December 2003.

[175] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *UC Berkeley, Dept. of Statistics, TR #649*, September 2003.

[176] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[177] S.G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics-Simulation and Computation*, 36:45–54, 2007.

[178] S.G. Walker, P.L. Damien, Purushottam W., and A.F.M. Smith. Bayesian nonparametric inference for random distributions and related functions. *Journal of the Royal Statistical Society, Series B*, 61(3):485–527, 1999.

[179] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

[180] L. Wasserman. Asymptotic properties of nonparametric Bayesian procedures. In D. Dey, P. M´uller, and D. Sinha, editors, *Practical Nonparametric and Semiparametric Bayesian Statistics*, pages 293–304. Springer-Verlag, 1998.

[181] Y. Weiss and W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[182] M. West. Bayesian factor regression models in the “large p, small n” paradigm. *Bayesian Statistics*, 7:723–732, 2003.

[183] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models.* Springer, 1997.

[184] R.A. Wijsman. On the attainment of the Cramér-Rao lower bound. *The Annals of Statistics*, pages 538–542, 1973.

[185] C. Wooters and M. Huijbregts. The ICSI RT07s speaker diarization system. *Lecture Notes in Computer Science*, 2007.

[186] E.P. Xing and K-A Sohn. Hidden Markov Dirichlet process: Modeling genetic inference in open ancestral space. *Bayesian Analysis*, 2(3):501–528, 2007.

[187] E.P. Xing, M.I. Jordan, and R. Sharan. Bayesian haplotype inference via the Dirichlet process. *Journal of Computational Biology*, 14(3):267–284, 2007.

[188] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proc. International Conference on Machine Learning*, June 2007.

[189] V.A. Yakubovich. Solutions of certain matrix inequalities in automatic control theory. *Dokl. Akad. Nauk USSR*, 143(3):1304–1307, 1962.