Hidden Process Models

Rebecca A. Hutchinson

CMU-CS-09-179

December 18, 2009

School of Computer Science
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Tom M. Mitchell, Chair
Zoubin Ghahramani
Marcel Just
Thomas Dean, Google

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the W.M. Keck Foundation under grant number DT123107; DARPA under grant number HR0011-04-1-0041; the National Science Foundation under grant number DGF-0254630; and the Space and Naval Warfare Systems Center, San Diego under grant number N66001-99-2-891808. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Hidden Process Models, probabilistic time series modeling, functional Magnetic Resonance Imaging, Expectation-Maximization, Dynamic Bayesian Networks, Markov Chain Monte Carlo, variational inference

For my parents,
and for Peter

Abstract

This thesis introduces Hidden Process Models (HPMs). HPMs are a probabilistic time series model for data assumed to be generated by a set of processes, where each process is characterized by a unique spatial-temporal signature and a probability distribution over its timing relative to a set of known timing landmarks. Research on HPMs has been inspired and motivated by the functional Magnetic Resonance Imaging (fMRI) domain, and this document presents, develops, and evaluates this framework in the context of fMRI. We provide the HPM formalism, inference and learning algorithms, extensions to the basic formalism, a discussion of the correspondence between HPMs and Dynamic Bayesian Networks, experimental results evaluating HPMs on real and synthetic fMRI data, and examples of how to visualize the learned models. We conclude that the HPM extensions incorporating domain knowledge about the process signatures are important for analyzing real fMRI data, and suggest future improvements to the model.

Acknowledgments

I'm very grateful to have had excellent mentors and collaborators over the course of my education. First, I want to thank my advisor, Tom Mitchell. I always left our meetings more optimistic than when I arrived, and your positivity was a huge source of support over the years. Thank you for believing in me even when I didn't.

I would like to thank my thesis committee, Tom Dean, Zoubin Ghahramani, and Marcel Just, for their feedback over the course of this research. Thanks also go to the members of the fMRI research group and CCBI over the years for many helpful discussions, including Kai-min Chang, Vladimir Cherkassky, Tim Keller, Vicente Malave, Rob Mason, Mark Palatucci, Svetlana Shinkareva, and Leonid Teverovskiy. In particular, Indra Rustandi collaborated on early stages of this research, and Francisco Periera helped me learn how to be a grad student, in both technical and non-technical realms.

There are many faculty members at CMU who made time for me at various steps along the way, but I especially want to thank Geoff Gordon for helpful conversations about this work. I also want to thank Sharon Cavlovich for making my life easier on many occasions.

I had several mentors as an undergraduate who helped me along the path to graduate school; thank you to Jerry Mead, Amy Greenwald, Mary Jean Harrold, Todd Griffith, and Steve Guattery.

Finally, thanks to Tom Dietterich for letting me start my post-doc early, and for being patient and supportive as I finished up.

I also want to thank my family and friends, whose support has sustained me over the years. Many friends have helped me in many ways, and I appreciate you all. In particular, I want to thank Sara for being a great lab partner, teammate, and friend. Thanks for letting me lean on you, and I'm sorry I didn't get a chance to name an algorithm after you. Thanks also to many friends for being good listeners, notably Kathy, Katy, Julie, Deb, Rachel, and Phoebe. I have to thank all of the exercise buddies that helped me blow off steam, too: my ultimate frisbee teammates in Pittsburgh (especially the ladies on Pounce),

my racquetball partners, and the group in Corvallis. To Caryn, Jeff, Brian, Christine, Ian, and Laura, thanks for helping me relax when I needed to. I also had great support from great officemates, Lucian and Brian.

I especially want to thank my sister, Ashley, and my parents, Kevin and Donna. Ash, I love having you for a sister and I appreciate all your support. Dad, you have been my academic role model and inspiration. Thanks for the encouragement and understanding along the way, and thanks for teaching me to write. Mom, I'm proud to have you as such a strong female role model in my life, balancing a successful career with family. Thanks for helping pave my way, and thanks for helping me with my math homework.

Finally, I want to thank my husband, Peter, who was with me through it all. Thanks for getting me through the lows, and helping me celebrate the highs. I don't know what I would've done without your support.

I feel lucky to have wonderful family and friends; love and thanks to all of you.

Contents

1 Introduction **1**

1.1 Motivation: functional Magnetic Resonance Imaging 1

1.2 A Running Example 2

1.3 Overview of Related Work 4

1.4 Thesis Statement 7

2 Hidden Process Models: Formalism and Algorithms **9**

2.1 Formalism 9

2.2 HPM Algorithms 17

2.2.1 Inference 17

2.2.2 Learning from Fully Observed Data 18

2.2.3 Learning from Partially Observed Data 20

2.3 Conclusion 22

3 HPM Extensions **23**

3.1 Regularizing the Process Response Signatures 23

3.1.1 Temporal Smoothness 25

3.1.2 Spatial Smoothness 25

3.1.3 Spatial Sparsity 26

3.1.4 Spatial Priors 26

3.1.5 Hybrid Penalties 27

3.2 Basis Functions for the Process Response Signatures 27
3.3 Continuous HPMs 29
3.4 Summary 36

4 A Dynamic Bayesian Network Formulation of HPMs 37
4.1 Writing HPMs as DBNs 38
4.1.1 Variables of HPM-DBNs 40
4.1.2 Parameters and Distributions of HPM-DBNs 40
4.2 Modeling Differences and Extensions 44
4.3 Algorithms for HPM-DBNs 47
4.4 A MCMC Sampling Algorithm for HPM-DBNs 47
4.4.1 Sampling Chains in HPM-DBNs 48
4.4.2 Updating the HPM-DBN Parameters 51

5 Experimental Results 55
5.1 Synthetic Data 55
5.1.1 Data Generation 56
5.1.2 Estimation of the Hemodynamic Responses 59
5.1.3 Choosing the Number of Processes to Model 62
5.2 Sentence-Picture Verification fMRI Data 62
5.2.1 Models 64
5.2.2 Comparing HPMs 71
5.2.3 Visualizing Learned HPMs 86

6 Conclusions 95
6.1 Discussion of Results 95
6.2 Advice on Using HPMs 97
6.3 Contributions 98
6.4 Future Directions 99
6.5 Summary 100

A Notation **103**

B Derivation of the M step Update for W **107**

C Tutorial on Factorial Hidden Markov Models **109**

C.1 Exact Inference 112

C.1.1 Naive Inference 112

C.1.2 An Improvement 113

C.1.3 Exact Inference for HPMs 113

C.2 Variational Inference 114

C.2.1 The General Framework for Variational Inference 115

C.2.2 Mean Field Approximation for Factorial HMMs 116

C.2.3 Structured Approximation for Factorial HMMs 120

D Derivation of Sampling Distributions for HPM-DBNs **129**

D.1 Sampling b 131

D.2 Sampling A 132

D.3 Sampling W 134

D.4 Sampling Σ 136

E Variational Inference for HPM-DBNs **137**

F Visualizations of Learned HPM Processes **149**

F.1 Standard HPMs 149

F.2 Regularized HPMs 153

F.3 HPMs with Basis Functions 157

Bibliography **161**

List of Figures

1.1 Experiment structure for the sentence-picture verification task. In 20 trials, the sentence was presented first, and in 20 trials the picture was presented first. The timelines show that each stimulus was presented for 4 seconds, with 4 seconds of fixation (looking at a cross) between the stimuli. Trials were separated by 15 second rest periods. Buttons could be pressed any time during the second stimulus presentation indicating whether the sentence correctly described the picture (in the upper example it does, in the lower example it does not). Example stimuli are shown above the timeline. 3

2.1 Example HPM for synthetic 2-process, 2-voxel data. Each process has a duration d, possible offsets Ω and their probabilities Θ, and a response signature W over time (on the horizontal axis) and space (voxels $v1$ and $v2$). They are instantiated 4 times in the configuration c. The start time of a process instance i is its landmark λ plus its offset o. The predicted mean of the data is the sum of the contributions of each process instance at each time point. (The response signatures are contrived rather than realistic in order to more clearly show linearity.) 10

3.1 Graphical model for a continuous HPM with two trials of data and two processes. Shaded nodes are observed; square nodes are discrete, round nodes are continuous. All elements of the configurations except the o_{ci} are fixed in advance. θ parameterizes an offset distribution, ω parameterizes the process response signature, σ contains a noise value for every voxel, C_n is a random variable over configurations for trial n, and Y_n is a $T \times V$ data matrix for trial n. The set of configurations for each trial are the same in this example; in general they could be different. In some cases the trial subscript n is suppressed for simplicity (i.e. o_{ci} is really o_{nci} so that the random variables for the offsets in the two different trials are distinguishable). 33

4.1 Example HPM-DBN. Triangles represent binary input nodes, squares represent discrete process nodes, circles represent continuous output nodes. Shaded nodes are observed. Each process has an arrow from its corresponding input type for each possible offset in Ω. Notation is provided for time slice $t = 3$. 39

4.2 Example process ordering constraint. The bold arrows encode the depencedce of the PressButton chain on the Decide chain. 46

5.1 Noise-free hemodynamic response functions of the processes in the synthetic data. 56

5.2 Two-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial is the reverse. 57

5.3 Three-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, which starts at time 60, is the reverse. The second peak in each trial is higher than in Figure 5.2 because the three-process data includes activation for the Decide process. 58

5.4 A sequence of two trials of synthetic data for the 2-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, starting at time 60, is the reverse. 59

5.5 Comparison of the learned vs. true process response signatures in the two-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.2647. 60

5.6 A sequence of two trials of synthetic data for the 3-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, starting at time 60, is the reverse. 61

5.7 Comparison of the learned vs. true process response signatures in the synthetic three-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.4427. 61

5.8 Basis functions used to parameterize process response signatures in sentence-picture experiments. 72

5.9 Screen shot of an HPM viewer written in Matlab. Drop-down menus select from the following views: observed data (for any particular trial), observed data averaged over all trials, predicted data (for any particular trial), and process response signatures (for any particular process). Anatomical regions of interest (ROIs) can also be highlighted, as demonstrated here for the visual cortex. 87

5.10 Screen shot of an HPM viewer written in Matlab. The viewer displays a single z-slice at a time, and clicking on a voxel brings up a new plot of the temporal response for that voxel (which was averaged to produce the original plot). Drop-down menus select from the following views: observed data (for any particular trial), observed data averaged over all trials, predicted data (for any particular trial), and process response signatures (for any particular process). Anatomical regions of interest (ROIs) can also be highlighted. 88

5.11 The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5). 89

5.12 The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using regularized HPMs for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5). 90

5.13 The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using HPMs with basis functions for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5). . . . 91

C.1 (a) Graphical representation of a Hidden Markov Model. S_t is the value of the hidden state at time t; Y_t is the observed data at time t. (b) Graphical representation of a factorial Hidden Markov Model with $M = 3$ hidden Markov chains. $S_t^{(m)}$ is the value of the hidden state of the m^{th} chain at time t. This figure is from Ghahramani and Jordan [1997]. 110

C.2 Graphical representation of the approximating family of distributions for the mean field variational approximation for factorial HMMs. $S_t^{(m)}$ represents the state of the m^{th} chain at time t. In the mean field approximation, all state variables are decoupled. This figure is from Ghahramani and Jordan [1997]. 117

C.3 Graphical representation of the approximating family of distributions for the structured variational approximation for factorial HMMs. $S_t^{(m)}$ represents the state of the m^{th} chain at time t. In this approximation, the temporal structure of each chain is retained, but the chains are treated independently (the arrows from the state variables to the observed variables are broken). This figure is from Ghahramani and Jordan [1997]. 121

E.1 Example HPM-DBN with process ordering constraints for which the variational inference algorithm is developed. This is the graphical structure of the probability distribution P in the text. 138

E.2 The structured approximation Q to the true HPM-DBN (P) shown in Figure E.1. 139

F.1 Learned ViewPicture process in HPM-3-K learned from all trials and all voxels using standard HPMs. 150

F.2 Learned ReadSentence process in HPM-3-K learned from all trials and all voxels using standard HPMs. 151

F.3 Learned Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs. 152

F.4 Learned ViewPicture process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs. 154

F.5 Learned ReadSentence process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs. 155

F.6 Learned Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs. 156
F.7 Learned ViewPicture process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions. 158
F.8 Learned ReadSentence process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions. 159
F.9 Learned Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions. 160

List of Tables

2.1 Sample configurations for a trial in the sentence-picture verification experiment. S = ReadSentenlce, P = ViewPicture, and D = Decide. These configurations reflect the assumptions that the S and P processes begin exactly when their corresponding stimuli are presented (in this case, at images 1 and 17), that the first two process instances are S and P in either order (but two instances of the same type are not allowed), and that the third process instance is D (but we do not have any additional information about its timing beyond the offset values defined for Decide). If we wish to customize this set of configurations to a specific trial, we can remove the configurations in which the identities of the first two process instances do not match the sequence of the stimuli in that trial. 14

3.1 The set of configurations in continuous HPMs that corresponds to the set of discrete HPM configurations in Table 2.1. The o_{ci}s are hidden random variables governed by the distributions specified by their corresponding processes. 30

5.1 For each training set, the table shows the average (over 30 runs) test set log-likelihood of each of 3 HPMs (with 2, 3, and 4 processes) on each of 3 synthetic data sets (generated with 2, 3, and 4 processes). Each cell is reported as mean $\pm$ standard deviation. NOTE: All values in this table are $*10^5$. 63

5.2 Configurations for a test set trial under HPM-2-K. For supervised training, there is only one configuration for each training trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset. 65

5.3 Configurations for a test set trial using HPM-2-U. For supervised training where we assume the order of the stimuli is known, there will be only four configurations for each training trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset. 66

5.4 Configurations for a test set trial under HPM-3-K. The reaction time for this trial is 2.6 seconds, which corresponds to offset 5 for the Decide process, so all configurations with offsets greater than 5 for this process have been eliminated for this trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset. 68

5.5 Configurations for a test set trial using HPM-4-K. For this trial, the participant's reaction time was 1.75 seconds, so the landmark for PressButton (B) is 3 (the nearest preceding image). We have eliminated configurations in which Decide (D) begins after PressButton. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset. 70

5.6 Improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). HPM-N-U is an HPM with N processes with unknown timing for ReadSentence and ViewPicture (the processes may begin 0 or 1 images after the corresponding stimulus is presented). 75

5.7 For the baseline HPM with the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 76

5.8 For HPMs with spatial smoothness and temporal smoothness regularization penalties on the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 77

5.9 For HPMs parameterized with basis functions using the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 78

5.10 Percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). HPM-N-U is an HPM with N processes with unknown timing for ReadSentence and ViewPicture (the processes may begin 0 or 1 images after the corresponding stimulus is presented). 82

5.11 For the baseline HPM with the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 83

5.12 For HPMs with spatial and temporal smoothness regularization penalties on the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 84

5.13 For HPMs parameterized with basis functions using the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). 85

5.14 Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs. 92

5.15 Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized HPMs. 92

5.16 Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions. . . 92

A.1 Notation used in the formalism for HPMs, roughly in order of introduction. The last column indicates the size and/or domain of the variable, where appropriate. 104

A.2 Notation used in descriptions of the HPM algorithms. The last column indicates the size and/or domain of the variable, where appropriate. 105

Chapter 1

Introduction

This thesis is inspired by high-dimensional time series data that is assumed to be generated by a set of underlying processes whose timings are partially observed. We define a process to have a spatial-temporal signature and a probability distribution over the times when it can occur relative to a set of known timing landmarks in the data. We are interested in simultaneously estimating the signatures and timings of the processes. To this end, we have developed Hidden Process Models (HPMs), which we introduce, develop, and evaluate below.

In this chapter, we motivate our efforts with the functional Magnetic Resonance Imaging (fMRI) domain, discuss related work, and state our thesis.

1.1 Motivation: functional Magnetic Resonance Imaging

Functional Magnetic Resonance Imaging is a safe, non-invasive technology that can be used to image the brain. Since the spatial resolution of fMRI is on the order of millimeters, and the temporal resolution is on the order of seconds, sequences of brain images taken while a person in the scanner performs mental tasks can provide rich data for cognitive scientists.

fMRI does not measure neuronal activity directly; instead, it measures a corellate of neuronal activity based on blood flow. Since oxygenated and deoxygenated hemoglobin have different magnetic properties (oxygenated hemoglobin is diamagnetic and deoxygenated hemoglobin is paramagnetic), the fMRI scanner can detect changes in the relative amounts of these two molecules. When metabolic function in a brain location increases

(presumably in response to increased neural activity), more oxygenated hemoglobin is delivered to that location. The additional oxygenated hemoglobin along with its subsequent conversion to deoxygenated hemoglobin constitute an effect measurable by fMRI, called the Blood Oxygenation Level Dependent (BOLD) response or the hemodynamic response. The BOLD response is sluggish compared to the time scale of the corresponding neural activity; less than a second of neural activation can cause a BOLD response lasting several seconds. While this may seem a crude approximation of the neural response, many studies have shown that there is sufficient signal in fMRI data for distinguishing among various kinds of cognitive states (see the discussion of related work below for examples).

1.2 A Running Example

To make the introduction of HPMs more clear, we will use concrete examples based on the following experimental paradigm for sentence-picture verification. We have both real fMRI data (Keller et al. [2001]) and synthesized data for this paradigm which we will describe later in more detail.

In our datasets, participants were presented with a sequence of 40 trials. In half of the trials, participants were shown a picture (involving vertical arrangements of the symbols *, +, and $) for 4 seconds, followed by a blank screen for 4 seconds, followed by a sentence (e.g. "The star is above the plus.") for 4 seconds. Participants could press buttons indicating whether the sentence correctly described the picture at any time during the second stimulus presentation. The participants then rested for 15 seconds before the next trial began. In the other half of the trials the sentence was presented first and the picture second, using the same timing. Figure 1.1 diagrams the temporal structure of the trials.

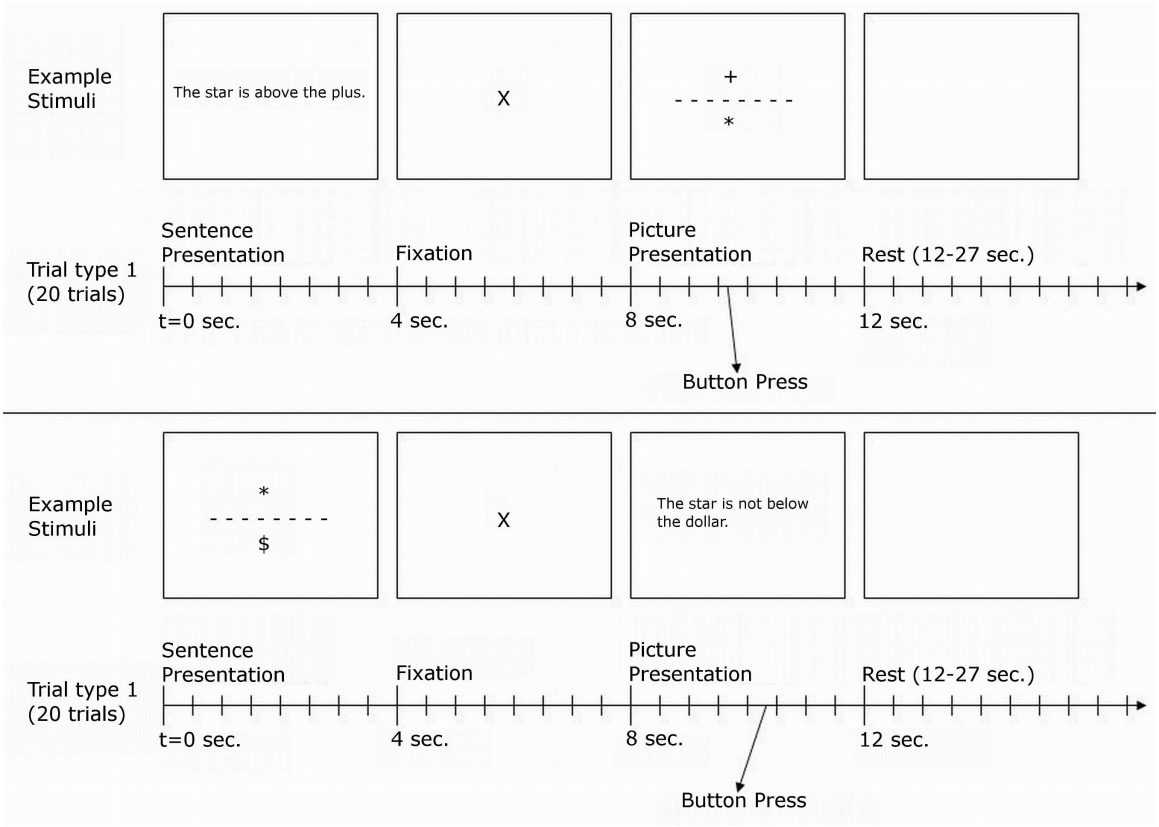

Figure 1.1: Experiment structure for the sentence-picture verification task. In 20 trials, the sentence was presented first, and in 20 trials the picture was presented first. The timelines show that each stimulus was presented for 4 seconds, with 4 seconds of fixation (looking at a cross) between the stimuli. Trials were separated by 15 second rest periods. Buttons could be pressed any time during the second stimulus presentation indicating whether the sentence correctly described the picture (in the upper example it does, in the lower example it does not). Example stimuli are shown above the timeline.

1.3 Overview of Related Work

Two major problems in analyzing fMRI data can be broadly described as detection and estimation. By detection, we mean the problem of determining which voxels in the brain are involved in the response to a particular stimulus or experimental condition. By estimation, we mean the problem of modeling the hemodynamic response function (HRF) for one or more stimuli or experimental conditions. The two problems are related, since detection is often done by comparing the observed fMRI data to the convolution of a hemodynamic response function with the time course of the experiment. Hidden Process Models address the problem of estimation.

Perhaps the most common approach to the problem of estimating the HRF is the General Linear Model (GLM), in which the data is modeled as a linear combination of the parameters of the HRF (to be estimated) and a design matrix describing the timing of the experiment. This method is detailed in Dale and Buckner [1997] and Dale [1999]. The work in Dale and Buckner [1997] helped push fMRI experiment design from blocked designs (in which stimuli are presented in rapid succession and the data is averaged over the presentations to improve the signal to noise ratio) to event-related designs (in which stimuli are presented and analyzed based on trials of different types) by showing that the linearity assumption of the GLM roughly holds for visual stimuli spaced as little as two seconds apart. This linearity allows the selective averaging of different trial types to estimate the HRFs. More mathematical detail on this model is given in Dale and Buckner [1997]. Previously, Boynton et al. [1996] also investigated the validity of the linearity assumption commonly made in fMRI analysis using blocked design experiments using blocks of different length presentations of visual stimuli. That work also found that the response to a longer stimulus can be reasonably approximated by summing and shifting the responses to shorter stimuli. Both Boynton et al. [1996] and Dale and Buckner [1997] find some deviations from linearity in their analyses, but both conclude that it is a reasonable working assumption for fMRI analysis. That said, researchers have also studied the nature of these deviations from linearity in fMRI, and developed alternative models of the HRF that account for nonlinearities. Friston et al. [2000] presents a nonlinear estimation technique for HRFs. Birn et al. [2001] discusses the spatial heterogeneity of the nonlinearities.

More recently, there have been several more sophisticated approaches to estimating hemodynamic response functions. Dale [1999] discusses an extension to the GLM in which the HRF parameters are modeled as weights on a set of basis functions, and Hossein-Zadeh et al. [2003] builds on that approach by showing how to choose the basis functions using Principal Components Analysis (PCA) (Jolliffe [2002]). The PCA is done over in-

stantiations of a canonical HRF form, the gamma function, that vary the peak-time and width parameters within reasonable ranges. They apply their method to the problem of detecting significantly active voxels and find that it is more sensitive than previous approaches using different subspaces to model the HRF. Another extension to the GLM is described in Ciuciu et al. [2003]. That work addresses the problem of estimating HRFs for events that are not synchronized with the imaging rate (asynchronous experiments), which can occur for instance for subject-initiated events with variable response times. The approach puts the whole regression on a finer temporal grid, on which both the image acquisition times and the events of interest are defined. The Bayesian estimation procedure for this model also imposes some constraints on the HRF through priors.

An effort to combine the detection and estimation problems in a single framework is described in Makni et al. [2008]. That paper uses an HRF model similar to the one in Ciuciu et al. [2003], and similar Bayesian estimation techniques. Makni et al. [2008] divides the brain into parcels that share an HRF shape, and uses an autoregressive noise model. They provide an algorithm for estimation of the HRF parameters and a method for making statistical comparisons across conditions in the framework. This represents a potential advance over treating detection and estimation sequentially, but it is still constrained to experimental conditions whose timings can reasonably be assumed known.

Another relevant field is that of mental chronometry, which is concerned with decomposing a cognitive task into its component processing stages. Traditionally, mental chronometry has relied on behavioral data such as measured reaction times, but studies have shown that functional MRI can also be used to address this problem (Menon et al. [1998], Formisano and Goebel [2003]). Menon et al. [1998] addresses the question of whether fMRI can be used to detect latency between brain areas. In a visual task, the authors find that the delay between presentations of stimuli in the left and right hemifields is highly correlated with the onsets of the HRFs for left and right areas of the visual cortex. In a motor task, they find that while the delay between the HRF onsets from the supplementary motor area to the main visual cortex is a constant of the task, the corresponding delay between the visual cortex and supplementary motor area scales linearly with reaction time. Since these types of studies rely on the relative onsets of HRFs (potentially for varying stimuli and/or brain areas), they clearly depend on accurate methods for estimating the delay or latency of the HRF from an event of interest. Two approaches for this problem are found in Henson et al. [2002] and Liao et al. [2002].

Returning to the problem of detecting which voxels are involved in an fMRI experiment, we must discuss the widely used Statistical Parametric Mapping (SPM) software (Friston [2003]). SPM convolves a canonical HRF with an on-off timecourse of fMRI stimuli and performs statistical hypothesis tests comparing the resulting time series with

the observed data. An advantage of SPM is that it produces maps describing the activation throughout the brain in response to particular stimuli. A disadvantage of SPM is that it is massively univariate, performing independent statistical tests for each voxel. In fact, it has been shown that some mental states cannot be detected with univariate techniques, but require multivariate analysis instead (Kamitani and Tong [2005]).

Two other time series analysis methods for detecting activity in fMRI are Lindquist et al. [2007] and Faisan et al. [2007]. Lindquist et al. [2007] uses techniques from statistical control theory and change-point theory to develop a Hierarchical Exponentially Weighted Moving Average (HEWMA) model that is particularly well-suited to experiments on phenomena that cannot be easily repeated, like studying the effects of a short-acting drug. This model detects whether a time series has shifted away from its baseline mean, and if so, when the shifts occurred. Faisan et al. [2007] presents hidden Markov multiple event sequence models (HMMESMs). HMMESMs use data that has been preprocessed into a series of spikes for each voxel, which are candidates for association with hemodynamic events. HMMESMs use an optimized, but fixed activation lag.

The field of Machine Learning has also had an impact on fMRI analysis. The use of classification techniques from the field of machine learning has become widespread in the fMRI domain (see Haynes and Rees [2006] for an overview). Classifiers have been used to predict group membership for particular participants (e.g. drug-addict vs. control, in Zhang et al. [2005]), and a variety of mental states (Mitchell et al. [2008], Palatucci and Mitchell [2007], Kamitani and Tong [2005], Mitchell et al. [2004], Cox and Savoy [2003], Haxby et al. [2001]). These classifiers assume that the mental processes being classified do not overlap in time, and that their timings are fully known during both training and testing. Kay et al. [2008] describes work on the identification of natural images by estimating a receptive field model for voxels in the visual cortex. After the model is trained on one set of images, brain activity is predicted for a novel set of test images and the image with the highest correlation with each observed trial is selected.

Dynamic Bayesian Networks (DBNs) are another class of models widely used in machine learning for time series analysis. The historical record of DBNs can be traced back to Dean and Kanazawa [1988] and Dean and Wellman [1991], with more recent contributions in Murphy [2002]. Examples of DBNs include Hidden Markov models (HMMs) (Rabiner [1989]) and factorial Hidden Markov Models (fHMMs) (Ghahramani and Jordan [1997]), which are particularly relevant to Hidden Process Models.

Finally, this thesis is partially based on work on HPMs reported in Hutchinson et al. [2006] and Hutchinson et al. [2009]. Portions of Hutchinson et al. [2009] are expanded upon in Niculescu [2005] and Niculescu et al. [2006] which use a simple version of HPMs as a case study for learning Bayesian networks with parameter constraints. In particular,

Niculescu's work showed that clustering neighboring voxels into groups that share parameters can improve HPM performance.

1.4 Thesis Statement

The thesis of the research reported here is that we can develop machine learning techniques for time series data that can simultaneously estimate parameters of temporal signatures and the onsets of the latent processes that generate them. These techniques can be improved by incorporating domain knowledge, and they can be used to express, learn, and compare competing models of the processes generating the data. This thesis is motivated by, and will be explored within the fMRI domain.

To support this thesis, we provide results on synthetic and real fMRI data using Hidden Process Models. The synthetic data results show that HPMs can recover the parameters of the model used to generate the data in an ideal situation, and that we can use held-out data log-likelihood to identify the model with the correct number of latent processes. The real data results indicate that the standard HPM parameterization and algorithms can overfit when faced with the complexity of fMRI data, but that extensions incorporating domain knowledge, like the temporal smoothness of the process response signatures, can overcome this problem. We also demonstrate how HPMs trained on fMRI data can be compared using cross-validated log-likelihood, and how they can be visualized for exploratory data analysis.

To our knowledge, HPMs are the first probabilistic model for fMRI data that can estimate the hemodynamic response for overlapping mental processes with unknown onset while simultaneously estimating a distribution over the timing of the processes. We hope that this research can help pave the way for a new paradigm of fMRI experiments that can explore sets of cognitive processes more complex than those that directly correspond to stimuli.

The rest of this document is organized as follows. In Chapter 2, we introduce Hidden Process Models, including the formalism for the model along with inference and learning algorithms. In Chapter 3, we present three extensions to the basic HPM presented in Chapter 2. In Chapter 4, we detail the correspondance between HPMs and Dynamic Bayesian Networks (DBNs), including indications of how one could adapt DBN algorithms for the specifics of HPMs. Chapter 5 presents experimental results on the performance of a variety of HPMs from Chapters 2 and 3 on real and synthetic fMRI datasets. Finally, Chapter 6 discusses the results from Chapter 5, suggests directions for future work, and concludes.

Chapter 2

Hidden Process Models: Formalism and Algorithms

In this chapter, we present the basic Hidden Process Model. We begin with a formal statement of Hidden Process Models. We then present algorithms for doing inference with HPMs, and algorithms for learning the HPM parameters in fully observable and partially observable settings.

2.1 Formalism

A Hidden Process Model is a description of a probability distribution over a time series, represented in terms of a set of processes, and a specification of their instantiations. HPMs assume the observed time series data is generated probabilistically by a latent collection of process instances. Each process instance is active during some interval of time, and influences the observed data only during this interval. Process instances inherit parameters from general process descriptions. The timing of a process instance depends on timing parameters of the general process it instantiates, plus a fixed timing landmark specific to the process instance. If multiple process instances are simultaneously active at any point in time, then their contributions sum linearly to determine their joint influence on the observed data. Figure 2.1 depicts an example of an HPM for synthetic fMRI data. We also supply a table of notation (Table A.1) to aid the reader.

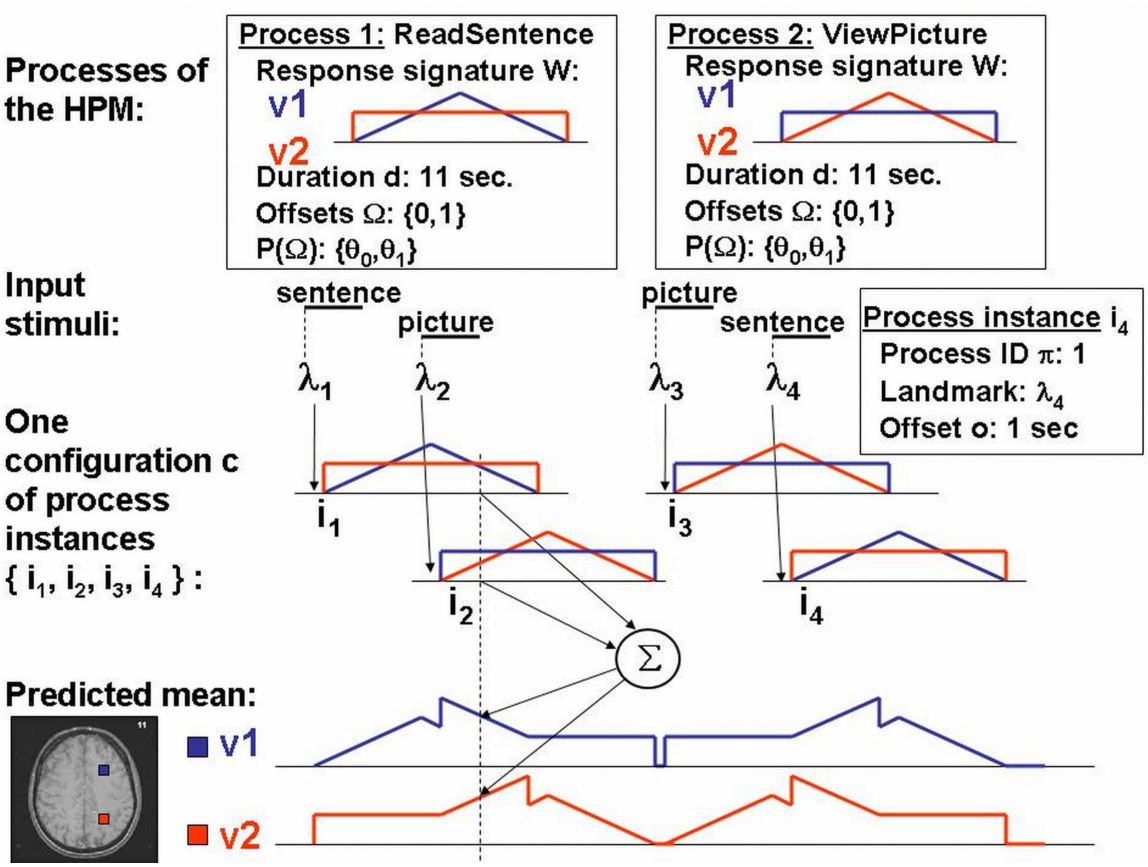

Figure 2.1: Example HPM for synthetic 2-process, 2-voxel data. Each process has a duration d, possible offsets Ω and their probabilities Θ, and a response signature W over time (on the horizontal axis) and space (voxels $v1$ and $v2$). They are instantiated 4 times in the configuration c. The start time of a process instance i is its landmark λ plus its offset o. The predicted mean of the data is the sum of the contributions of each process instance at each time point. (The response signatures are contrived rather than realistic in order to more clearly show linearity.)

More formally, we consider the problem setting in which we are given observed data $\mathbf{Y}$, a $T \times V$ matrix consisting of V time series, each of length T. For example, these may be the time series of fMRI activation at V different voxels in the brain. The observed data $\mathbf{Y}$ is assumed to be generated nondeterministically by some system. We use an HPM to model this system. Let us begin by defining processes:

Definition 1 *A process π is a tuple $\langle \mathbf{W}, \Theta, \Omega, d \rangle$. d is a scalar called the duration of π, which specifies the length of the interval during which π is active. $\mathbf{W}$ is a $d \times V$ matrix called the response signature of π, which specifies the influence of π on the observed data at each of d time points, in each of the V observed time series. Θ is a vector of parameters that defines a multinomial distribution over a discrete-valued random variable which governs the timing of instances of π, and which takes on values from the set Ω of integers. The set of all processes is denoted by Π.*

We will use the notation $\Omega(\pi)$ to refer to the parameter Ω for a particular process π. More generally, we adopt the convention that $f(x)$ refers to the parameter f affiliated with entity x.

Each process represents a general procedure which may be instantiated multiple times over the time series. For example, in the sentence-picture fMRI study described above, we can hypothesize general cognitive processes such as ReadSentence, ViewPicture, and Decide. While it is reasonable to assume that the ReadSentence and ViewPicture processes begin exactly when their corresponding stimuli are presented (in this case, at images 1 and 17 in the time series), we must treat the onset time for Decide with less certainty. A reasonable assumption might be that Decide begins within 4 seconds after the second stimulus presentation. To reflect these assumptions, we can define the processes ReadSentence and ViewPicture as both having their sets of possible onsets relative to the stimulus presentation as $\Omega = \{0\}$, and Decide as having $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7\}$. To model the hemodynamic response, we might choose a 12 second interval, which corresponds to $d = 24$ time points. $\Omega(\pi)$ and $d(\pi)$ are modelling choices, and the values of these parameters for all processes are specified a priori in the HPM. The parameters $\mathbf{W}(\pi)$ and $\Theta(\pi)$ $\forall \pi$ can be specified a priori or learned from data.

For the sentence-picture example, each process would be instantiated once for each trial. The instantiation of a process at a particular time is called a *process instance*, defined as follows:

Definition 2 *A process instance i is a tuple $\langle \pi, \lambda, O \rangle$, where π identifies a process as defined above, λ is a known scalar called a timing landmark that refers to a particular*

point in the time series, and O is an integer random variable called the offset, which takes on values in $\Omega(\pi)$. The time at which process instance i begins is defined to be $\lambda + O$. The multinomial distribution governing O is defined by $\Theta(\pi)$. The duration of i is given by $d(\pi)$, and the response signature is $\mathbf{W}(\pi)$.

The timing landmark λ is a time point in the data, typically associated with an event in the experiment design. For example, the timing landmark for a process instance in the sentence-picture verification task could be the time at which a particular stimulus is presented. $\lambda(i)$ specifies roughly when process instance i will begin, subject to some variability depending on its offset $O(i)$, which in turn depends on its process idendity $\pi(i)$. More specifically, the process instance i starts at $t = \lambda(i) + O(i)$, where $O(i)$ is a random variable whose distribution is a property of the process $\pi(i)$. In contrast, the particular value of $O(i)$ is a property of the instance. That is, while there may be some variation in the offset times of different ReadSentence process instances, we assume that the amount of time between each sentence stimulus (the timing landmark λ) and the beginning of its corresponding ReadSentence process instance follows the same distribution, as specified in the ReadSentence process.

We consider process instances that may generate the data in ordered sets, rather than as individual entities. This allows us to use knowledge of the experiment design to constrain the model. We refer to each possible set of process instances as a *configuration*.

Definition 3 *A configuration c of length I is a set of process instances $\{i_1 \dots i_I\}$, in which all parameters for all instances ($\{\lambda(i), \pi(i), O(i)\}, i = \{1 \dots I\}$) are fully-specified. The set of all configurations is given by $\mathcal{C}$.*

Each configuration is an assignment of particular values to the variables $\{\lambda(i), \pi(i), O(i)\}$ for some number of process instances I. The purpose of the set of configurations $\mathcal{C}$ is to reduce the hypothesis space of the HPM in a reasonable way. We can think of the hypothesis space of an HPM as the number of ways that its processes can be instantiated to influence the data, which is the set of all possible configurations. In general, if any process can be instantiated at any time, the hypothesis space can be very large. To see this, consider that each process instance has T possible landmarks, $|\Pi|$ possible process IDs, and $|\Omega(\pi)|$ possible offsets once π is chosen. The choice of process ID and offset are not independent, but they are jointly independent from the choice of landmark, resulting in $T \sum_\pi |\Omega(\pi)|$ possibilities for each process instance. If we have I process instances, they can be combined in $(T \sum_\pi |\Omega(\pi)|)^I$ ways. The set $\mathcal{C}$ restricts the hypothesis space of the model by allowing the HPM to consider a much smaller number of possibilities. These

configurations also facilitate the incorporation of prior knowledge about the experiment design.

For example, in the sentence-picture verification study, if we use the processes ReadSentence, ViewPicture, and Decide as discussed above, we can define the set of configurations given in Table 2.1. These configurations reflect the assumptions that the ReadSentence and ViewPicture processes begin exactly when their corresponding stimuli are presented (in this case, at images 1 and 17), that the first two process instances are ReadSentence and ViewPicture in either order (but two instances of the same type are not allowed), and that the third process instance is Decide (but we do not have any additional information about its timing beyond the offset values defined for Decide). This set of configurations is for a general trial. To customize it for a particular trial, we can remove the configurations in which the identities of the first two process instances do not match the sequence of the stimuli in that trial. As written, Table 2.1 restricts us to 16 of a potential $(54(1+1+8))^3 = 1.57464 * 10^8$ configurations of length 3.

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$	$\pi(i_3)$	$\lambda(i_3)$	$O(i_3)$
1	S	1	0	P	17	0	D	17	0
2	S	1	0	P	17	0	D	17	1
3	S	1	0	P	17	0	D	17	2
4	S	1	0	P	17	0	D	17	3
5	S	1	0	P	17	0	D	17	4
6	S	1	0	P	17	0	D	17	5
7	S	1	0	P	17	0	D	17	6
8	S	1	0	P	17	0	D	17	7
9	P	1	0	S	17	0	D	17	0
10	P	1	0	S	17	0	D	17	1
11	P	1	0	S	17	0	D	17	2
12	P	1	0	S	17	0	D	17	3
13	P	1	0	S	17	0	D	17	4
14	P	1	0	S	17	0	D	17	5
15	P	1	0	S	17	0	D	17	6
16	P	1	0	S	17	0	D	17	7

Table 2.1: Sample configurations for a trial in the sentence-picture verification experiment. S = ReadSentenlce, P = ViewPicture, and D = Decide. These configurations reflect the assumptions that the S and P processes begin exactly when their corresponding stimuli are presented (in this case, at images 1 and 17), that the first two process instances are S and P in either order (but two instances of the same type are not allowed), and that the third process instance is D (but we do not have any additional information about its timing beyond the offset values defined for Decide). If we wish to customize this set of configurations to a specific trial, we can remove the configurations in which the identities of the first two process instances do not match the sequence of the stimuli in that trial.

We can now define Hidden Process Models:

Definition 4 *A Hidden Process Model,* HPM*, is a tuple* $\langle \Pi, \mathcal{C}, \langle \sigma_1^2 \ldots \sigma_V^2 \rangle \rangle$*, where* Π *is a set of processes,* $\mathcal{C}$ *is a set of configurations, and* σ_v^2 *is the variance characterizing the Gaussian distribution governing the* v^{th} *time series of* $\mathbf{Y}$*. An* HPM *defines a probability distribution over the observed data* $\mathbf{Y}$ *as follows:*

$$P(\mathbf{Y}|\text{HPM}) = \sum_{c \in \mathcal{C}} P(\mathbf{Y}|\text{HPM}, C = c) P(C = c|\text{HPM}) \tag{2.1}$$

where $\mathcal{C}$ *is the set of configurations associated with the* HPM*, and* C *is a random variable defined over* $\mathcal{C}$*.*

The term $P(\mathbf{Y}|\text{HPM}, C = c)$ *is:*

$$P(\mathbf{Y}|\text{HPM}, C = c) = \frac{1}{(2\pi)^{\frac{VT}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (Y - \mu(c))' \mathbf{\Sigma}^{-1} (Y - \mu(c)) \right) \tag{2.2}$$

where Y *and* $\mu(c)$ *are column vectors of length* VT*,* $\mu(c)$ *represents the mean of the Gaussian defined by configuration* c*, and* $\mathbf{\Sigma}$ *is a* $VT \times VT$ *matrix whose diagonal is* T *repetitions of* σ_1^2*, followed by* T *repetitions of* σ_2^2*, and so on through* σ_V^2*.*

The term $P(C = c|\text{HPM})$ *defines a prior distribution over the set of configurations.*

In other words, $P(\mathbf{Y}|HPM)$ is a mixture of Gaussians, with each mixture component generated by a particular configuration. Note that the set of configurations C is defined as part of the HPM and that it is fixed in advance.

To understand this definition, first consider a single configuration $c = \{i_1 \ldots i_I\}$ and a single observed data point y_{tv}. The probability distribution over y_{tv} is defined by the Gaussian distribution:

$$P(y_{tv}|c, HPM) \sim \mathcal{N}(\mu_{tv}(c), \sigma_v^2) \tag{2.3}$$

where σ_v^2 is the variance characterizing the time-independent, voxel-dependent noise distribution associated with the v^{th} time series, and where

$$\mu_{tv}(c) = \sum_{i \in c} \sum_{\tau=0}^{d(\pi(i))} \delta(\lambda(i) + O(i) = t - \tau)\, \mathbf{W}_{\tau v}(\pi(i)) \tag{2.4}$$

Here $\delta(\cdot)$ is an indicator function whose value is 1 if its argument is true, and 0 otherwise. $\mathbf{W}_{\tau v}(\pi(i))$ is the element of the response signature $\mathbf{W}$ associated with process $\pi(i)$, for data series v, and for the τ^{th} time step in the interval during which i is instantiated.

Equation 2.4 says that for a given configuration c, the mean of the Gaussian distribution governing observed data point y_{tv} is the sum of single contributions from each process instance whose interval of influence includes time t. In particular, the $\delta(\cdot)$ expression is non-zero only when the start time ($\lambda(i) + O(i)$) of process instance i is exactly τ time steps before t, in which case we add the element of the response signature $\mathbf{W}_v(\pi(i))$ at the appropriate delay (τ) to the mean at time t. This expression captures a linear system assumption that if multiple processes are simultaneously active, their contributions to the data sum linearly. (To some extent, this assumption holds for fMRI data (Boynton et al. [1996]) and is widely used in fMRI data analysis.)

Alternatively, we can define $\mu_{tv}(c)$ using a design matrix created from c called $\mathbf{X_c}$, which is $T \times D$, where $D = \sum_\pi d(\pi)$. The t^{th} row of $\mathbf{X_c}$ is non-zero only in the columns corresponding to the time points of the process response signatures that influence that time point. If responses overlap temporally, there will be more than one non-zero element in the row. If two process instances of the same process begin at the same time, $\mathbf{X_c}$ will be 2 for the time points where the two process instances occur. Essentially, $\mathbf{X_c}$ maps the time points of the process response signatures onto the time points of the observed data according to the configuration c. Using this approach, we can write:

$$\mu(c) = \mathbf{X_c}\mathbf{W} \tag{2.5}$$

where $\mathbf{W}$ ($D \times V$) refers to the vertical concatentation of the $\mathbf{W}(\pi)\ \forall\pi$, and $\mu(c)$ is $T \times V$.

The second term in Equation 2.1 is the prior distribution over the set of configurations. One choice for this distribution is:

$$\begin{aligned} P(C = c|\textit{HPM}) &\propto P(\bigwedge_{i \in c} \pi(i) \wedge O(i)|\textit{HPM}) \\ &\propto P(\bigwedge_{i \in c} \pi(i)|\textit{HPM}) \prod_{i \in c} P(O(i)|\pi(i), \textit{HPM}) \\ &\propto P(\bigwedge_{i \in c} \pi(i)|\textit{HPM}) \prod_{i \in c} \Theta_{O(i)}(\pi(i)) \end{aligned} \tag{2.6}$$

In the basic version of HPMs, we define the joint probability $P(\bigwedge_{i \in c} \pi(i)|\textit{HPM})$ to be a uniform distribution over all possible combinations of process IDs.

Thus, the generative model for an *HPM* involves first choosing a configuration $c \in \mathcal{C}$, using the distribution given by Equation 2.6, then generating values for each time point t of each time series v using the configuration c of process instances and the distribution for $P(\mathbf{Y}|\textit{HPM}, C = c)$ given by Equations 2.3 and 2.4.

Note that if T is large, the number of configurations C can be very large even for a modest amount of uncertainty. In some cases, we can store fewer configurations by exploiting conditional independencies in the structure of the data. fMRI data is grouped into N trials, which are typically designed to allow all the hemodynamic responses in one trial to decay before beginning a new trial. In domains other than fMRI, the notion of trials translates to windows of data separated by periods during which no process instances are active under any configuration. This structure essentially creates conditional independencies in the model, since $P(Y_n|c)$ only depends on the process instances of c that are active during trial n. These process instances can be separated into N independent sets of configurations $\mathcal{C}_n$. Equation 2.6 can be applied to $\mathcal{C}_n$:

$$P(C_n = c_n|\mathit{HPM}) = P(\bigwedge_{i \in c_n} \pi(i)|\mathit{HPM}) \prod_{i \in c_n} \Theta_{O(i)}(\pi(i))$$

Then the probability of a configuration over all trials is:

$$P(C = \{c_1 \ldots c_n\}|\mathbf{Y}, \mathit{HPM}) = \prod_{n=1}^{N} P(C_n = c_n|\mathbf{Y_n}, \mathit{HPM}) \tag{2.7}$$

We will use this notion of trial-structured data to explain the inference algorithm below, but in general HPMs can describe data with or without this structure.

2.2 HPM Algorithms

In this section we present algorithms for doing inference and learning in HPMs. The basic inference problem in HPMs is to infer the posterior distribution $P(C = c|\mathbf{Y}, \mathit{HPM})$ over the configurations $\mathcal{C}$, given the *HPM* and observed data $\mathbf{Y}$. The learning problem in HPMs is to learn maximum likelihood estimates of the HPM parameters, given an observed data sequence $\mathbf{Y}$ and a set of configurations $\mathcal{C}$. Let $\Psi = \{\Theta(\pi), \mathbf{W}(\pi), \sigma_v^2\} \forall \pi, \forall v$ be the set of HPM parameters to be learned.

2.2.1 Inference

We can infer the posterior distribution $P(C = c|\mathbf{Y}, \mathit{HPM})$ over the configurations $\mathcal{C}$, given the *HPM* and observed data $\mathbf{Y}$ with a straightforward application of Bayes theorem. First, the $P(C_n = c_n|\mathbf{Y_n}, \mathit{HPM})$ within trial n is:

$$P(C_n = c_n|\mathbf{Y_n}, \mathit{HPM}) = \frac{P(\mathbf{Y_n}|C_n = c_n, \mathit{HPM}) P(C_n = c_n|\mathit{HPM})}{\sum_{c'_n \in \mathcal{C}_n} P(\mathbf{Y_n}|C_n = c'_n, \mathit{HPM}) P(C_n = c'_n|\mathit{HPM})} \tag{2.8}$$

where $\mathbf{Y_n}$ is the data for trial n, and where the terms in this expression can be obtained using Equations 2.3, 2.4, and 2.6. Over multiple trials we have:

$$P(C = \{c_1 \ldots c_n\}|\mathbf{Y}, \textit{HPM}) = \prod_{n=1}^{N} P(C_n = c_n|\mathbf{Y_n}, \textit{HPM}) \tag{2.9}$$

Given the posterior probabilities over the configurations, we can easily compute marginal probabilities to answer more general questions. For instance, we can compute the probability of the identity of a process instance by summing the probabilities of the configurations in which the process instance in question takes on the identity of interest. More concretely, suppose we wish to know the probability that the second process instance i_2 in trial 4 has identity S. This quantity is given by:

$$P(C_4(\pi(i_2)) = S) = \sum_{c \in \mathcal{C}_4} \delta(c((\pi(i_2)) = S))P(C_4 = c|\mathbf{Y_4}, \textit{HPM}) \tag{2.10}$$

where again, $\delta(\cdot)$ is an indicator function that returns 1 if and only if its argument is true.

Note that other marginal probabilities can be obtained similarly from the posterior distribution, such as the probabilities of particular offsets for the process instances, or the joint probability of two process instances having a particular pair of identities.

2.2.2 Learning from Fully Observed Data

First consider the case in which the true configuration of process instances is fully observed in advance (i.e. there is only one configuration in the HPM). For example, in our sentence-picture verification experiment, we might assume that there are only two processes, ReadSentence and ViewPicture, that each process begins exactly when its corresponding stimulus does ($\Omega = \{0\}$), and that the observed sequence of stimuli corresponds exactly to the sequence of process instance IDs, resulting in a single configuration.

In such fully observable settings the problem of learning $\Theta(\pi)$ reduces to a simple maximum likelihood estimate of multinomial parameters from observed data:

$$\hat{\Theta}_o(\pi) \longleftarrow \frac{\sum_{i \in c} \delta(\pi(i) = \pi \wedge O(i) = o)}{\sum_{i \in c, o' \in \Omega(\pi(i))} \delta(\pi(i) = \pi \wedge O(i) = o')} \tag{2.11}$$

Alternatively, we can use an m-estimate (Mitchell [1997]) to update the estimate of $\Theta(\pi)$:

$$\hat{\Theta}_o(\pi) \longleftarrow \frac{[\sum_{i \in c} \delta(\pi(i) = \pi \wedge O(i) = o)] + mp}{[\sum_{i \in c, o' \in \Omega(\pi(i))} \delta(\pi(i) = \pi \wedge O(i) = o')] + m} \tag{2.12}$$

Here p is the prior on $\Theta_o(\pi)$ (e.g. $p = \frac{1}{|O(\pi)|}$) and the parameter m controls the weight of the prior as compared to the training data. This method avoids setting some of the $\Theta(\pi)$ parameters to 0 when their corresponding offsets are not observed in the training data, since we may wish to allow that offset with non-zero probability in the test set.

The problem of learning the response signatures $\mathbf{W}(\pi)$ is slightly more complex, because the $\mathbf{W}(\pi)$ terms from multiple process instances can jointly influence the observed data at each time point (see equation (2.4)). Solving for $\mathbf{W}$ reduces to solving a multiple linear regression problem to find a least squares solution. Our multiple linear regression approach in this case is based on the General Linear Model approach for estimating hemodynamic responses in fMRI data as described in Dale [1999].

Since there is only one configuration c of process instances in this setting, it is simple to create $\mathbf{X_c}$ as described above (see Equation 2.5). We define $\tilde{\mathbf{X}}_\mathbf{c}$ to be a $VT \times VD$ matrix whose block-diagonal is the $T \times D$ matrix $\mathbf{X_c}$ repeated V times. (The off-diagonal blocks are all zero, since they would essentially map the process response signatures for one voxel onto the data for another voxel.) A summary of the notation used in this section is in Table A.2.

The linear system we are modeling is then:

$$Y = \tilde{\mathbf{X}}_\mathbf{c} W + \epsilon \tag{2.13}$$

where $\epsilon \sim N(0, \sigma^2)$ and σ^2 refers to a $VT \times 1$ vector where σ_v^2 is tiled appropriately. Minimizing the objective function:

$$\begin{aligned} f_0(W) &= ||Y - \tilde{\mathbf{X}}_\mathbf{c} W||^2_{\sigma^{-1}} \\ &= (Y - \tilde{\mathbf{X}}_\mathbf{c} W)' \mathbf{\Sigma}^{-1} (Y - \tilde{\mathbf{X}}_\mathbf{c} W) \end{aligned} \tag{2.14}$$

where $\mathbf{\Sigma}$ has σ^2 on the diagonal, gives:

$$\hat{\mathbf{W}} \longleftarrow (\tilde{\mathbf{X}}_\mathbf{c}{}' \mathbf{\Sigma}^{-1} \tilde{\mathbf{X}}_\mathbf{c})^{-1} \tilde{\mathbf{X}}_\mathbf{c}{}' \mathbf{\Sigma}^{-1} Y \tag{2.15}$$

One complication that arises is that the regression problem can be ill posed if the training data does not exhibit sufficient diversity in the relative onset times of different process instances. For example, if processes A and B always occur simultaneously with the same onset times, then it is impossible to distinguish their relative contributions to the observed data. In cases where the problem involves such singularities, we use the Moore-Penrose pseudoinverse to solve the regression problem.

Having computed the maximum likelihood estimates for $\mathbf{W}$, it is easy to find the maximum likelihood solution for the σ_v^2:

$$\hat{\sigma_v^2} \longleftarrow \frac{1}{T}\sum_{t=1}^{T}(y_{tv} - \hat{\mu}_{tv}(c))^2 \tag{2.16}$$

where $\hat{\mu}(\mathbf{c}) = \tilde{\mathbf{X}_\mathbf{c}}\hat{W}$.

2.2.3 Learning from Partially Observed Data

In the more general case, we may not know which of the set of configurations $\mathcal{C}$ is correct, and we face a problem of learning from incomplete data. The configurations can have different numbers of process instances, and different values for the parameters $\{\pi(i), \lambda(i), O(i)\}\forall i$. In trial-structured data, the configurations $\mathcal{C}_n$ can have different lengths for different n if we have more uncertainty about some trials than others. If we have no knowledge of the experiment design at all, we can list all possible combinations of process instances. If we do have some knowledge, it can be incorporated into the set of configurations as discussed above.

In the presence of uncertainty over configurations, we use an Expectation-Maximization (EM) algorithm (Dempster et al. [1977]) to estimate the parameters Ψ. Let us call the latent variable of interest Z, a binary vector of length C indicating which of the configurations is correct, where $\sum_c Z_c = 1$. The EM algorithm locally maximizes the expected log-likelihood of the complete (observed and unobserved) data, which we call Q:

$$Q(\Psi, \Psi^{\text{old}}) = E_{Z|\mathbf{Y},\Psi^{\text{old}}}[\ln P(\mathbf{Y}, Z|\Psi)] \tag{2.17}$$

The EM algorithm finds parameters Ψ that locally maximize the Q function by iterating over the following two steps until the change in Q from one iteration to the next drops below a threshold, or until a maximum number of iterations is reached:

E step: Solve for $E_{Z|\mathbf{Y},\Psi^{\text{old}}}[Z]$, where $E_{Z|\mathbf{Y},\Psi^{\text{old}}}[Z_c] = P(C = c|\mathbf{Y}, \Psi^{\text{old}})$ is the probability of configuration c given $\mathbf{Y}$ and Ψ^{old}. The solution to this is given by Equation (2.9).

M step: Use $E_{Z|\mathbf{Y},\Psi^{\text{old}}}[Z]$ from the E step to obtain new parameter estimates for Ψ that maximize Q (Equation 2.17).

Below, we use $E[Z]$ for $E_{Z|\mathbf{Y},\Psi^{\text{old}}}[Z]$ to simplify notation. We additionally introduce a design matrix covering all configurations $\tilde{\mathbf{X}}$, and corresponding parameters $\tilde{Y}$, $\tilde{Z}$, and

$\tilde{\mathbf{\Sigma}}$. Let $M = V \sum_n T_n C_n$ be the number of rows in the design matrix $\tilde{\mathbf{X}}$ representing all possible configurations for all trials. The other variables in our linear system need to be tiled or repeated in order to match the size of this design matrix, $\tilde{\mathbf{X}}$.

$\tilde{\mathbf{X}}$ is constructed in the following way. First we create $\tilde{\mathbf{X}_\mathbf{n}}$ for all trials n by vertically concatenating the $\tilde{\mathbf{X}_{\mathbf{nc}}}$ as defined above for all $c \in n$. Then $\tilde{\mathbf{X}}$ is the vertical concatenation of the $\tilde{\mathbf{X}_\mathbf{n}}$ for all n. $\tilde{\mathbf{X}_{\mathbf{nc}}}$ is $VT_n \times VD$, $\tilde{\mathbf{X}_\mathbf{n}}$ is $VT_nC_n \times VD$, and $\tilde{\mathbf{X}}$ is $M \times VD$.

$\tilde{Y}$ is the $M \times 1$ vector of data, created by vertically concatenating the copies of Y_n for every configuration in trial n, for every trial. $\tilde{Z}$ is a $M \times 1$ vector in which the Z_{nc} for each configuration is repeated for the rows of $\tilde{\mathbf{X}}$ to which it corresponds. We also define $\tilde{\mathbf{Z}} = diag(\tilde{Z})$. Similarly, $\tilde{\mathbf{\Sigma}}$ is an $M \times M$ covariance matrix in which σ_v^2 is repeated for every element of M that refers to voxel v. $\tilde{\sigma}^2$ is the diagonal of $\tilde{\mathbf{\Sigma}}$. Finally, we define $\mathbf{\Upsilon} = \tilde{\mathbf{\Sigma}}^{-1} E[\tilde{\mathbf{Z}}]$.

Using this notation, we maximize Q by minimizing:

$$f_0(W) = E[\tilde{Z}]||\tilde{Y} - \tilde{\mathbf{X}}W||^2_{\tilde{\sigma}^{-1}} \tag{2.18}$$

The solution to this weighted least squares problem is:

$$\hat{W} \longleftarrow \left(\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} \tag{2.19}$$

The full derivation for Equation 2.19 is given in Appendix B. Intuitively, this update simply extends the least squares solution for observed data (Equation 2.15) by tiling the design matrix so that it includes all possible configurations, and adding weights to the optimization problem that reflect the probability of each configuration.

Given the update for $\hat{\mathbf{W}}$, the updates for the other parameters in Ψ are:

$$\hat{\sigma}_v^2 \longleftarrow \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{c_n \in \mathcal{C}_n} E[z_{nc}](y_{ntv} - \mu_{ntv}(c_n))^2 \tag{2.20}$$

$$\hat{\theta}_{\pi, O=o} \longleftarrow \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{c_n \in \mathcal{C}_n, i \in c_n} \delta(\pi(i) = \pi \wedge O(i) = o) E[z_{nc}]}{\sum_{c'_n \in \mathcal{C}, i' \in c'_n, o' \in \Omega(\pi(i'))} \delta(\pi(i) = \pi \wedge O(i) = o') E[z_{nc'}]} \tag{2.21}$$

While this solution for updating $\mathbf{W}$ is correct, it is computationally intensive. For typical fMRI data, the value of $M = V \sum_n T_n C_n$ (the number of rows in the design matrix for the entire experiment) can be very large. In our sentence-picture verification example, $V \approx 5000$, and $T_n \approx 54$. The number of configurations per trial C_n reflects our uncertainty about the processes occuring and is therefore highly variable. For the

reasonable assumptions represented in Table 2.1, $C_n = 16$ for each of the 40 trials. This yields $M = 5000 * 40 * 54 * 16 \approx 1.73 * 10^8$. To model a 12 second hemodynamic response for each of 3 processes, we would use $D = \sum_\pi d(\pi) = 24 * 3 = 72$ time points for the response signatures. Therefore, the $M \times VD$ design matrix $\tilde{\mathbf{X}}$ would be $1.73 * 10^8 \times 72$.

In practice, we can reduce M significantly by requiring all the rows of $\tilde{\mathbf{X}}$ to be unique. Each row of $\tilde{\mathbf{X}}$ corresponds to a time point in the experiment, and it is easy to see that multiple configurations can 'share' a particular explanation of a time point. For instance, the first 2 configurations in Table 2.1 agree on the explanations for time points 1 through 16, and only begin to differ at time point 17 on which time point of the D process influences the data. The other variables in the least squares solution for $\mathbf{W}$ can be scaled similarly to match the smaller design matrix.

2.3 Conclusion

In this chapter, we have presented the basic Hidden Process Model, detailing the formalism of the model and several algorithms. We showed how to do inference over a set of configurations of process instances using Bayes rule. We explained how to use the General Linear Model to estimate the parameters of the model when we know the correct configuration of process instances. Along with the supplemental material in Appendix B, we derived an extension of the GLM to estimate the HPM parameters under uncertainty about the correct configuration. In the following chapters, we will explore alternative parameterizations of the process response signatures and process offsets, and an alternative formalism based on Dynamic Bayes Networks.

Chapter 3

HPM Extensions

In this chapter, we explore three extensions to the basic HPM formalism presented above. The first extension is regularization for the process response signatures, which incorporates bias into the HPM learning algorithm to direct the parameters toward signatures with desirable properties (e.g. temporal smoothness) without committing to a parametric form for the responses. The second extension models the process response signatures using weights on a set of basis functions, reducing the number of free parameters by restricting the model to the pre-specified subspace defined by the basis. Both of these extensions require only simple modifications to the learning algorithm presented in Chapter 2. The third extension allows the hidden offset variables to be continuous, effectively removing the constraint that processes must start on the temporal grid of the data acquisition rate. Relaxing this assumption also requires we commit to a continuous parametic form for the process response signatures. The continuous version of HPMs is also sufficiently more complex than the HPMs in Chapter 2 that we introduce a sampling algorithm in place of the EM algorithm described above.

3.1 Regularizing the Process Response Signatures

The parameterization of the process response signatures as matrices of independently estimated weights to be summed in predicting the output of an HPM has advantages and disadvantages. The main advantage is flexibility; this parameterization can capture a wide variety of functional forms. In comparison, a model that commits a priori to a particular form of the response (e.g. a canonical HRF) risks the possibility that the form may be incorrect. The main disadvantage of the nonparametric approach is the large number of

parameters to be estimated, especially as compared to the relatively sparse amount of data available. In some experiment designs, this parameterization also suffers from identifiability issues, where there may be an infinite number of solutions to the parameter estimation problem that produce the same data log-likelihood. There is clearly a spectrum from most to least modeling flexibility, and this parameterization will often be too flexible to produce interesting results.

One way to address this problem without committing to a (potentially incorrect) parametric form for the process response signatures is to add regularization penalties to the learning algorithm. These penalties bias the parameter estimates toward biologically plausible properties. We can do this by adding a term $g(W)$ to the objective function that penalizes deviations from these properties. This penalty term is weighted by another parameter γ which specifies how much importance should be given to the penalty term relative to the optimization term involving the data. The general form of the objective function we minimized to derive the algorithms above (from Appendix B), with regularization penalty $g(W)$, is:

$$\begin{aligned} f_0(W) &= E[\tilde{Z}]||\tilde{Y} - \tilde{\mathbf{X}}W||^2_{\tilde{\sigma}^{-1}} + \gamma g(W) \\ &= (\tilde{Y} - \tilde{\mathbf{X}}W)'\tilde{\mathbf{\Sigma}}^{-1}E[\tilde{\mathbf{Z}}](\tilde{Y} - \tilde{\mathbf{X}}W) + \gamma g(W) \end{aligned} \tag{3.1}$$

High values for γ increase the bias toward results with low penalty terms; low values for γ allow the fit to the data to be more influential.

Notice that since the we are simply adding the regularization term to the objective function we had before, we can simply add the derivative of the penalty term to the derivative we had before, set the new derivative equal to 0, and solve for W:

$$\frac{df_0}{dW} = -2\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} + 2\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}W + \gamma\frac{dg}{dW} \tag{3.2}$$

For the fMRI domain, there are several biologically plausible properties for which we may want to regularize. The rest of this section suggests penalty terms to regularize for temporal smoothness, spatial smoothness, spatial sparsity, and spatial priors. In each case, we will write penalties in terms of the matrices $\mathbf{W}(\pi)$, each of which is $d(\pi) \times V$, or time by space. We will continue to vectorize these matrices for the parameter estimation formulae above, but the penalties are easier to interpret in terms of the individual matrices.

3.1.1 Temporal Smoothness

Since the HRF reflects a measurement of blood flow dynamics in the brain, we would like our estimates to be temporally smooth. To impose this constraint on the $\mathbf{W}(\pi)$, we can use a squared $\mathcal{L}_2$ penalty on the differences between successive time points for each voxel and each process:

$$g_1(W) = \sum_{\pi \in \Pi} \sum_{j=1}^{V} \sum_{i=1}^{d(\pi)-1} (\mathbf{W}(\pi)_{j,i+1} - \mathbf{W}(\pi)_{j,i})^2 \tag{3.3}$$

The derivative with respect to $\mathbf{W}$ is:

$$\begin{aligned} \frac{dg_1}{d\mathbf{W}(p)_{v,a}} &= 2(\mathbf{W}(p)_{v,a} - \mathbf{W}(p)_{v,a-1}) - 2(\mathbf{W}(p)_{v,a+1} - \mathbf{W}(p)_{v,a}) \\ &= 4\mathbf{W}(p)_{v,a} - 2\mathbf{W}(p)_{v,a-1} - 2\mathbf{W}(p)_{v,a+1} \end{aligned} \tag{3.4}$$

for $1 < a < d(p)$ (the boundary cases each have only one of the $a+1$ or $a-1$ terms).

3.1.2 Spatial Smoothness

Since voxels that are close to each other will have similar blood flow dynamics, we would like the estimates of the HRFs in these voxels to be similar. To penalize deviations from spatial smoothness, we first assume we have a binary adjacency matrix $\mathbf{A}$ that is $V \times V$, where $\mathbf{A}(i,j) = 1$ if and only if voxel i is adjacent to voxel j for some definition of 'adjacent.' Suppose we wish to impose spatial smoothness on the estimates of $\mathbf{W}$, meaning that if 2 voxels are adjacent, we wish to penalize deviations between the timecourses of their process response signatures. Again we can use a squared $\mathcal{L}_2$ norm, but this time the norm is over differences between spatially adjacent timecourses:

$$g_2(W) = \sum_{\pi \in \Pi} \sum_{i=1}^{V-1} \sum_{j=i+1}^{V} \mathbf{A}(i,j) \sum_{k=1}^{d(\pi)} (\mathbf{W}(\pi)_{i,k} - \mathbf{W}(\pi)_{j,k})^2 \tag{3.5}$$

The derivative with respect to $\mathbf{W}$ is:

$$\frac{dg_2}{d\mathbf{W}(p)_{v,a}} = 2 \sum_{j=v+1}^{V} \mathbf{A}(v,j)(\mathbf{W}(p)_{v,a} - \mathbf{W}(p)_{j,a}) - 2 \sum_{i=1}^{v-1} \mathbf{A}(i,v)(\mathbf{W}(p)_{i,a} - \mathbf{W}(p)_{v,a}) \tag{3.6}$$

for $1 < v < V$ (again the boundary cases have fewer terms).

3.1.3 Spatial Sparsity

By spatial sparsity, we mean that we want a small number of voxels to be active for any given process, but we do not want the time course of the active voxels to be sparse. That is, we want a sparsity penalty on the columns of $\mathbf{W}$ but not on the rows. One way to achieve this penalty is with a (2,1)-norm (Argyriou et al. [2008] uses this norm for a similar problem in multi-task learning):

$$g_3(W) = \sum_{\pi \in \Pi} \sum_{i=1}^{V} |\sqrt{\sum_{j=1}^{d(\pi)} \mathbf{W}(\pi)_{i,j}^2}| \tag{3.7}$$

The derivative with respect to W is:

$$\frac{dg_3}{d\mathbf{W}(p)_{v,a}} = \frac{\mathbf{W}(p)_{v,a} sgn(\sqrt{\sum_{j=1}^{d(p)} \mathbf{W}(p)_{v,j}^2})}{\sqrt{\sum_{j=1}^{d(p)} \mathbf{W}(p)_{v,j}^2}} \tag{3.8}$$

where

$$sgn(x) = \frac{x}{|x|} \tag{3.9}$$

3.1.4 Spatial Priors

In some cases, we may have prior knowledge about the regions of the brain that will be most active during a particular cognitive process. For example, we expect that visual stimuli will produce a response in the visual cortex, but minimal activity in other parts of the brain. In this situation, we can encode our prior knowledge in a $\mathbf{W}^{prior}$. We can then use regularization to penalize deviations from our expectations, again with a squared $\mathcal{L}_2$ norm:

$$g_4(W) = \sum_{\pi \in \Pi} \sum_{i=1}^{V} \sum_{j=1}^{d(\pi)} (\mathbf{W}(\pi)_{i,j} - \mathbf{W}(\pi)_{i,j}^{prior})^2 \tag{3.10}$$

$$\frac{dg_4}{d\mathbf{W}(p)_{v,a}} = 2(\mathbf{W}(p)_{v,a} - \mathbf{W}(p)_{v,a}^{prior}) \tag{3.11}$$

3.1.5 Hybrid Penalties

Any of the previous penalties can be combined with arbitrary weights to introduce multiple biases simultaneously:

$$\begin{aligned} g(W) &= \sum_i \gamma_i g_i(W) \\ \frac{dg}{dW} &= \sum_i \gamma_i \frac{dg_i}{dW} \end{aligned} \tag{3.12}$$

where the values of the γ_i reflect the relative importance of each penalty.

In Chapter 5, we present experimental results showing that regularizing the process weights for spatial and temporal smoothness improves upon the performance of the unregularized models.

While these regularization schemes avoid committing to particular forms of the response signatures, they still require a large number of parameters. In addition, another step must be added to the estimation process to choose the regularization weights (the γ values), though these can be hand-tuned and/or fixed in advance. In the next section, we describe a way to reduce the number of parameters in the model while sacrificing only a modest amount of flexibility.

3.2 Basis Functions for the Process Response Signatures

In this section, we introduce a parameterization of the process response signatures using a set of basis functions. We do not address the problem of learning a good basis set; rather, we assume the set of basis functions is fixed in advance. As we will see, this parameterization can reduce the number of parameters to be estimated and, if the basis is chosen carefully, restrict the model to a subspace with desirable properties.

Instead of using independent parameters for the time points of process response signatures, we can represent the contribution of each process to the observed output as a linear combination of a set of basis functions. The new parameters of the process response signatures are a set of B coefficients for each voxel, with each coefficient corresponding to one of B basis functions. We represent the basis functions themselves in a pre-defined matrix $\mathbf{H}$, which is $d \times B$, where $\mathbf{H}_{db}$ is the value of basis function b at time step d. We can then replicate this matrix along the block diagonal of a new matrix $\tilde{\mathbf{H}}$, which will be $VD \times BPV$, where P is the number of processes. We represent the coefficients as a

$BPV \times 1$ vector β. (Here, we assume that all processes have the same duration d, which is also the length of the basis functions.) Then the $VD \times 1$ vector W can be written as:

$$W = \tilde{\mathbf{H}}\beta \tag{3.13}$$

This simple change translates easily to the M step update derived in Appendix B, with the update for β now given by:

$$\hat{\beta} \leftarrow (\tilde{\mathbf{H}}'\tilde{\mathbf{X}}'\boldsymbol{\Upsilon}\tilde{\mathbf{X}}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}'\tilde{\mathbf{X}}'\boldsymbol{\Upsilon}\tilde{Y} \tag{3.14}$$

There are two main advantages to this reparameterization of the response signatures. Firstly, by choosing smooth basis functions we can impose temporal smoothness on the process responses and predicted means. We can also force the beginning and end of the responses to tend to zero by choosing the basis functions to begin and end near zero. Secondly, if we choose the number of basis functions B to be smaller than the number of independent time points d in W, we can reduce the number of parameters in the HPM. Where previously each process required d parameters for each voxel (for each time point of the response), using basis functions requires only B parameters for each voxel (coefficients for each basis function). For example, in the sentence-picture verification study where we acquired 2 images per second, we would need $D = 24$ to model a 12 second hemodynamic response function. If instead we use 3 basis functions to model this experiment, we can estimate $\frac{1}{8}$ of the number of parameters in the previous model.

The obvious remaining question is how to choose the number and form of the basis functions. In general, choosing the optimal basis set for fMRI data is outside the scope of this thesis. The model can generally be expected to perform well with basis functions if the basis can represent the true process response signatures; if the basis cannot represent the true model then the nonparametric approach presented above (especially the regularized version) may be a better option. One approach to modeling the HRF with basis functions is presented in Hossein-Zadeh et al. [2003]. That work suggests choosing a plausible functional form for the HRF with some number of parameters (they use a gamma function with two parameters). The HRF is then instantiated many times, for varying settings of the parameters within some reasonable range. They construct a $P \times N$ matrix $\mathbf{Q}$ consisting of P realizations of the HRF using different parameter settings, evaluated at N time points. They perform PCA on $\mathbf{Q}$, and use the first M principal components as basis functions (they use $M = 3$). This approach satisfies two general criteria to consider when choosing a basis set for HPMs: that the basis be able to represent a relevant class of functions (i.e. believed to contain the process response signatures), and that the basis be small enough to reduce the number of parameters in the model (i.e. fewer than d basis functions).

In practice, HPMs parameterized with basis functions generated using the Hossein-Zadeh et al. [2003] approach performed similarly to regularized HPMs, but took less time to train. Experimental results on these models are presented in Chapter 5.

3.3 Continuous HPMs

In the HPM formalism presented so far, the implicit assumption has been that processes begin at time points on the temporal grid defined by the data acquisition rate. This constraint results from the discrete parameterization of both the offset values and the process response signatures. The time points of the process response signatures are assumed to correspond to time points in the discrete data, and the offsets are integers added to landmarks that are also defined on the temporal grid of the data. Landmarks that occur between data points must be rounded to the nearest data point to align the process response signatures correctly. In fMRI, the assumption that cognitive processes begin on the same temporal grid as the image acquisitions is clearly false. In this section, we seek to eliminate this assumption and allow process start times (and therefore process offset values) to vary continuously.

In the model presented in Chapter 2, which we will refer to as the 'discrete HPM' below, the legal discrete offset values for each process π were listed in the vector $\Omega(\pi)$, and their corresponding multinomial parameters were in $\Theta(\pi)$. To allow continuous process start times, we must allow continuous offset values for the processes. The offset values for a process could either be bounded by a pre-specified closed interval $[l(\pi), h(\pi)]$, or they could be unbounded. In the first case, one possible modeling choice is a Beta distribution over that interval, with parameters $\alpha(\pi)$ and $\beta(\pi)$ taking the place of $\Theta(\pi)$. The second case is even more general, and one example of an appropriate probability distribution here is a Gaussian with parameters $\mu(\pi)$ and $\sigma^2(\pi)$. In either case, each process defines a distribution $P(o(i)|\pi(i); \Theta(\pi))$ where $\Theta(\pi)$ is used generally to denote process-specific parameters of whatever distribution is being used (multinomial, Beta, Gaussian, etc.).

This reparameterization of offsets requires a redefinition of the notion of configurations. Whereas configurations in the discrete model enumerated a process ID, landmark, and offset for each process instance, configurations in the continuous model only enumerate values for the process ID and landmark, which are then combined with a hidden, continuous offset random variable. For example, the hypothesis space defined by the configurations listed in Table 2.1 for the discrete model corresponds to the configurations shown in Table 3.1 for the continuous model. We write o_{ci} to mean the hidden offset variable for process instance i under configuration c. Note that we have only two configu-

rations here as compared to 16 in the discrete case.

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$	$\pi(i_3)$	$\lambda(i_3)$	$O(i_3)$
1	S	1	o_{11}	P	17	o_{12}	D	17	o_{13}
2	P	1	o_{21}	S	17	o_{22}	D	17	o_{23}

Table 3.1: The set of configurations in continuous HPMs that corresponds to the set of discrete HPM configurations in Table 2.1. The o_{ci}s are hidden random variables governed by the distributions specified by their corresponding processes.

These parameterized configurations introduce another modeling difference between discrete and continuous HPMs. While both models have a set of configurations to explicitly define the hypothesis space of the model, there are some types of process ordering constraints that can be represented with the discrete HPM configurations but not with the parameterized continuous HPM configurations. To see this, consider two process instances that we wish to constrain to a particular ordering, even if their intervals of legal start times ($\lambda + o$) overlap. This might be the case for processes like Decide and PressButton in the sentence-picture verification example; both of these processes should begin shortly after the second stimulus is presented, but Decide should begin before PressButton. In constructing configurations for a discrete HPM, we enumerate all possible combinations of the offsets, and we can then eliminate any configurations containing combinations with an illegal ordering. On the other hand, continuous HPMs introduce random variables for each of these offsets, and they are drawn independently of one another, from the distributions specified by their corresponding processes. There is no constraint to say that one offset value must be larger or smaller than another. A modeling tradeoff then, is that continuous HPMs allow start times to be removed from the temporal grid of the observations at the expense of the ability to specify strict process orderings in the hypothesis space for some cases.

Continuous offset values and start times also imply the need for continuous process response signatures, since we will need to be able to evaluate the response signatures at the times of the observed data points, regardless of how those time points relate to the continuous start time. Essentially, we need a continuous parametric form for the process response signature, which we will call $h(x; \omega(\pi))$, whose contribution to an observed data point at time t is $h(t - s; \omega(\pi))$. Here s is the continuous start time of an instance of process π, and $\omega(\pi)$ contains process-specific parameters of h. $h(x; \omega(\pi))$ could have a pre-specified duration d as in the discrete HPMOne example of a possible form for $h(x; \omega(\pi))$ is a gamma function with 3 parameters, where the parameters can vary across

voxels ($h(x;\omega(\pi,v))$):

$$h(x;\omega(\pi)) = \nu * \frac{(\frac{x}{\rho})^{\eta-1}\exp(-\frac{x}{\rho})}{\rho(\eta-1)!} \tag{3.15}$$

where ν, ρ, and η are defined for each process and each voxel ($\{\nu(\pi,v),\rho(\pi,v),\eta(\pi,v)\}$). ν controls the amplitude of the response, ρ controls the width of the peak, and η controls the delay of the peak. (This is the functional form of the impulse response used in Boynton et al. [1996].)

So far, we have purposefully left the specific forms of the continuous distribution over offsets and the process response signatures unspecified because the framework of continuous HPMs is sufficiently general to allow for a variety of modeling choices in these areas. The general form of the joint distribution specified by continuous HPMs can be written as:

$$P(O,C,Y) = P(O)P(C|O)P(Y|C,O)$$

where O represents the set of all hidden offset variables, C represents a random variable defined over configurations, and Y is the observed data.

We now further specify that the individual offset variables are marginally independent:

$$P(O) = \prod_{c\in\mathcal{C}}\prod_{i\in c} P(o_{ci}|\pi(i)) \tag{3.16}$$

where $P(o_{ci}|\pi(i))$ is the same as the $P(o(i)|\pi(i);\Theta(\pi))$ defined above for each process but with the parameters suppressed for brevity (i.e. this could be a Beta or Gaussian distribution depending on modeling choices).

Similarly to discrete HPMs, we specify the probability of the configurations as being proportional to the likelihood of their components:

$$P(C=c|O) \propto P(\wedge_{i\in c}\pi(i))\prod_{i\in c} P(o_{ci}|\pi(i)) \tag{3.17}$$

As before, we will treat the joint distribution over process IDs as uniform, essentially leaving us with:

$$P(C=c|O) \propto \prod_{i\in c} P(o_{ci}|\pi(i)) \tag{3.18}$$

Finally, the distribution over the observed data given the hidden variables remains Gaussian:

$$P(Y|C=c,O=o) = \prod_{v=1}^{V}\prod_{n=1}^{N}\frac{T(n)}{\sqrt{2\pi\sigma_v^2}}\prod_{t=1}^{T(n)}\exp\left(-\frac{1}{2\sigma_v^2}(y_{tv}-\mu_{tv}(c,o))^2\right) \tag{3.19}$$

where

$$\mu_{tv}(c, o) = \sum_{i \in c} h(t - \lambda(i) - o_{ci}; \omega(\pi(i), v)) \tag{3.20}$$

The generative model corresponding to these equations works as follows. Given a set of parameterized configurations and a set of processes, we first draw values for the offset variables for every configuration. That is, for each process instance i in each configuration c, we draw o_{ci} using the process parameters indicated by the process ID for this instance ($\pi(i)$). Once the configurations are fully instantiated with offset values, we draw a value for the correct configuration c from a multinomial distribution whose parameters are proportional to the likelihood of the offsets in each configuration. Now that we have a particular configuration, we can predict the mean of the Gaussian over the observed data by summing the process responses of all the instances in c over time, again using the parameters indicated in the configuration for their process IDs. A graphical model showing a continuous 2-process HPM for two trials is shown in Figure 3.1.

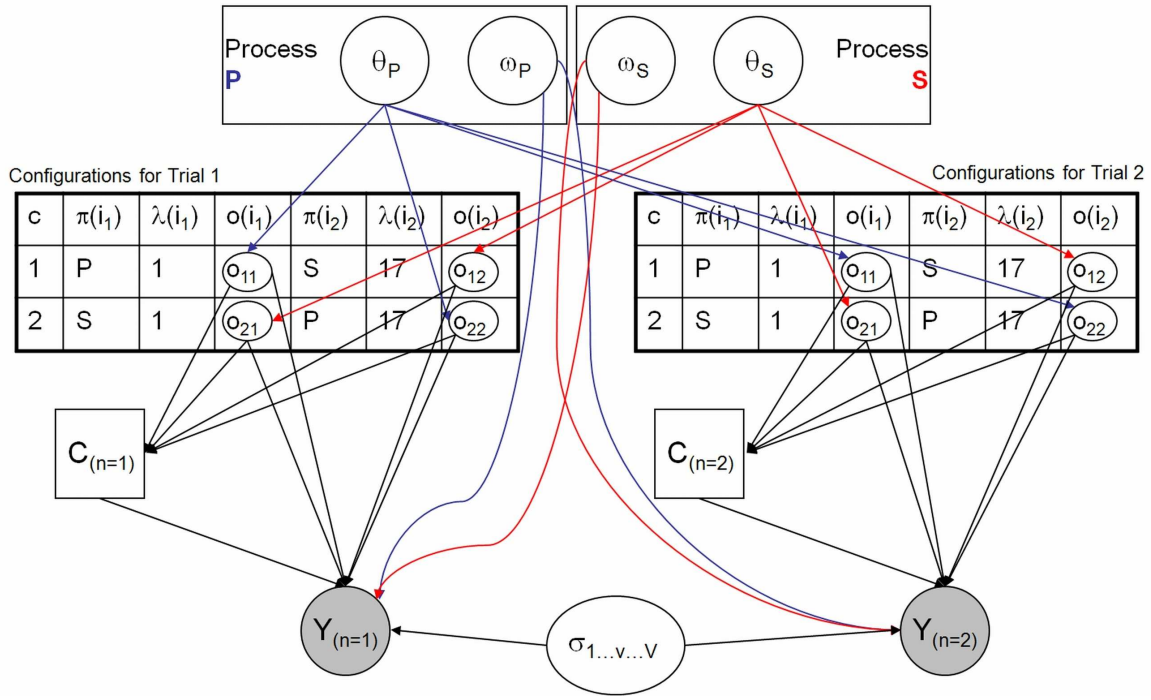

Figure 3.1: Graphical model for a continuous HPM with two trials of data and two processes. Shaded nodes are observed; square nodes are discrete, round nodes are continuous. All elements of the configurations except the o_{ci} are fixed in advance. θ parameterizes an offset distribution, ω parameterizes the process response signature, σ contains a noise value for every voxel, C_n is a random variable over configurations for trial n, and Y_n is a $T \times V$ data matrix for trial n. The set of configurations for each trial are the same in this example; in general they could be different. In some cases the trial subscript n is suppressed for simplicity (i.e. o_{ci} is really o_{nci} so that the random variables for the offsets in the two different trials are distinguishable).

Regardless of the specific distribution chosen to model $P(o_i|\pi(i);\theta(\pi))$ and the specific form of $h(x;\omega(\pi,v))$, the shift from discrete to continuous offsets introduces several intractable integrals into the EM derivation presented in Appendix B. We therefore introduce a different algorithm for inference and learning in continuous HPMs: Metropolis sampling. MacKay [2003] and MacKay [1998] provide excellent introductions to Markov Chain Monte Carlo (MCMC) methods, of which Metropolis sampling is one example.

The concept of the Metropolis algorithm is fairly simple. Suppose we are given a joint probability distribution $P(Y,X;\theta)$ over observed variables Y and hidden variables X. Furthermore, let us treat any unknown parameters θ the same as hidden variables (so $X = \{X,\theta\}$). Assuming that we can evaluate $P(Y,X)$ (or that we can evaluate it except for the normalization constant), we can draw samples from $P(X|Y)$ and use statistics of these samples to make statements about the values of the hidden variables and unknown parameters. The Metropolis algorithm starts from an initial value $X^{(0)}$ and proposes a change $X^{(prop)}$ from a proposal distribution Q. For example, Q might be a Gaussian distribution with fixed variance centered on the current point $X^{(0)}$. This new sample $X^{(prop)}$ is accepted with probability:

$$a = \min\left(1, \frac{P(Y,X^{(prop)})}{P(Y,X^{(0)})}\frac{Q(X^{(0)};X^{(prop)})}{Q(X^{(prop)};X^{(0)})}\right) \tag{3.21}$$

If the proposal distribution Q is symmetric (like a Gaussian), then $Q(X^{(0)};X^{(prop)}) = Q(X^{(prop)};X^{(0)})$ and this simplifies to:

$$a = \min\left(1, \frac{P(Y,X^{(prop)})}{P(Y,X^{(0)})}\right) \tag{3.22}$$

This equation says that if $X^{(prop)}$ increases the joint likelihood, it is accepted with probability 1. If it decreases the joint likelihood, the acceptance probability depends on the ratio of the likelihoods under the current and proposed values for X. If $X^{(prop)}$ is rejected, $X^{(0)}$ is added to the list of samples again. We continue to generate samples $X^{(t+1)}$ in this manner to acquire some number of samples. This process defines a Markov chain that is guaranteed to converge to the true distribution $P(X|Y)$ as $t \to \infty$.

More concretely, the hidden variables and parameters to be sampled in a continuous HPM are $X = \{\theta(\pi),\omega(\pi,v),\sigma^2(v),o_{nci},C(n)\}\forall\pi,v,n,c,i$. This assumes we are interested in learning the parameters; if the parameters have already been estimated or are assumed to be known instead, we can add them to the observed variables Y instead and sample only the hidden variables. Let us define a fully factorized proposal distribution Q consisting of an independent Gaussian distribution for each variable and parameter, each

with known variance (e.g. $Q(C(n)^{(prop)}) \sim N(C(n)^{(t-1)}, \sigma^2_{C(n)})$). The Metropolis algorithm proceeds as follows:

- Initialize $X^{(0)}$.
- For $t = 1 : T$
  - Generate $X^{(prop)}$ by sampling a new value for each element i in $X^{(t-1)}$ from that element's proposal distribution $Q(i)$.
  - Use Equation 3.16 (the joint likelihood) to compute a as in Equation 3.22 (the acceptance probability under a symmetric proposal density).
  - If $a = 1$, set $X^{(t)} = X^{(prop)}$.
  - Otherwise, set $X^{(t)} = X^{(prop)}$ with probability a and $X^{(t)} = X^{(t-1)}$ with probability $1 - a$.
- Disregard the first S samples (for which the chain has not yet converged to the true distribution).
- Compute statistics of interest (like mean and variance) about the hidden variables and/or parameters from the remaining $T - S$ samples.

The values of T and S depend on problem size.

The advantages of continuous HPMs are that they relax an assumption made in discrete HPMs that is obviously wrong: that process start times must coincide with the data acquisition rate. The disadvantages of this model, in addition to the inability to model some kinds of process orderings as discussed above, are the difficulties associated with Metropolis sampling. While Metropolis sampling is guaranteed to converge to the true distribution as $t \rightarrow \infty$, convergence can be slow because the hidden state space is explored with a random walk, and it can be hard to determine whether or not the chain has converged. Convergence can also be sensitive to the proposal distribution ($Q(X)$ above), the distribution over the latent variables and parameters ($P(X)$ above), and the initial values. Furthermore, since the samples are correlated, we need to collect many samples before they can be treated as effectively independent.

In practice, we have only achieved convergence with this algorithm on synthetic data for a toy problem (2 voxels). Even on the toy problem, convergence only occurred when the algorithm was initialized with parameters equal to (or very nearly equal to) the true values and the proposal distribution was sharply peaked around the true values. Since we do not expect to be able to recreate these conditions with real fMRI data, we have not applied MCMC to the sentence-picture study.

3.4 Summary

The goal of the extensions in this chapter is to provide a variety of modeling choices for HPMs. We showed how to add regularization to the learning algorithm to bias the process response signature parameters toward several desirable properties. We presented an alternative parameterization of the process response signatures using a set of basis functions that can reduce the number of parameters in the model while restricting the responses to a general class of smooth functions. Finally, we introduced continuous HPMs which allow the hidden processes to begin at any time, regardless of whether that time corresponds to an observed data point.

All of the extensions in this chapter retained the notion of configurations, which allows extreme flexibility in specifying the hypothesis space of the model. In the next chapter, we will examine a version of HPMs that does not use configurations.

Chapter 4

A Dynamic Bayesian Network Formulation of HPMs

The formalism for Hidden Process Models presented above allows a great deal of flexibility in terms of the hypothesis space of the model. By explicitly listing configurations in the HPM, the model can specify the number of process instances, the order of process instances, and interactions between the start times of process instances, simply by including or not including configurations that satisfy various constraints in the model. The cost of this flexibility is an inference algorithm whose complexity grows with the number of configurations, since every configuration must be considered explicitly to determine which one is most likely. If we have significant prior knowledge about how process instances should be arranged for a particular data set, then we can take advantage of that knowledge with a small number of configurations. However, as our uncertainty about the data set grows, we experience a computational explosion in the number of configurations.

In this chapter, we explore a fundamentally different way to encode prior knowledge in HPMs. We present an alternative modeling framework for HPMs that does not use configurations using Dynamic Bayesian Networks (DBNs) (Murphy [2002]), discuss the differences between this model and the previous ones. We also present algorithms for doing Bayesian inference and learning in this framework. We will refer to the new model as an HPM-DBN, to distinguish it from the HPMs presented above.

4.1 Writing HPMs as DBNs

In this section, we describe how to write a Hidden Process Model as a Dynamic Bayesian Network (DBN) (Murphy [2002]). Factorial Hidden Markov Models (fHMMs, Ghahramani and Jordan [1997]) are a subclass of DBNs that can express many of the assumptions of HPMs if we constrain them properly. (An introduction to fHMMs is provided in Appendix C for the unfamiliar reader.) In HPM-DBNs, we will have a Markov chain for every process we wish to model, but with restricted dynamics to reflect our assumptions about processes. Similarly to discrete HPMs, HPM-DBNs will define a Gaussian probability distribution over the observed data points, the mean of which will be a sum of contributions from each process-chain. We will also introduce input variables for each process-chain to model landmarks. We now proceed to the specifics and notations of HPM-DBNs. A diagram of an example HPM-DBN is provided in Figure 4.1 to supplement the text.

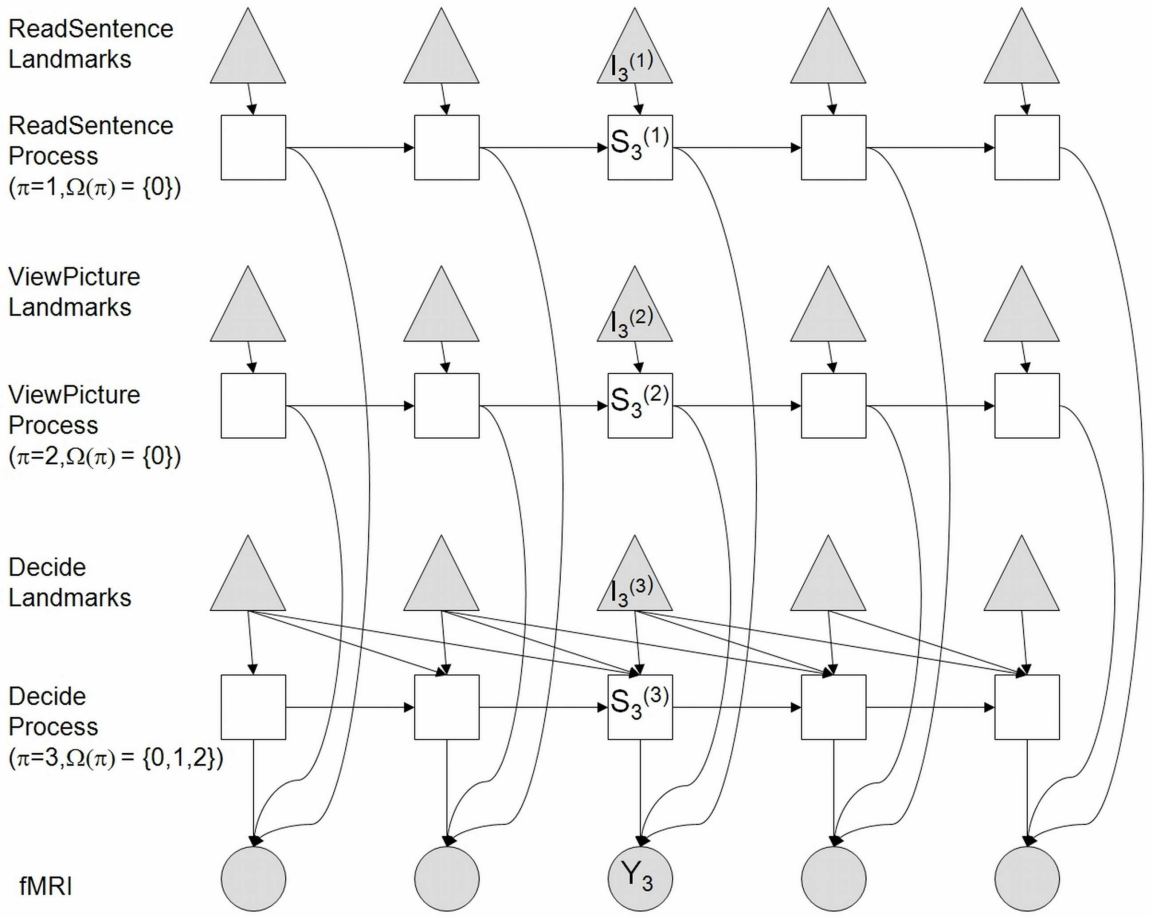

Figure 4.1: Example HPM-DBN. Triangles represent binary input nodes, squares represent discrete process nodes, circles represent continuous output nodes. Shaded nodes are observed. Each process has an arrow from its corresponding input type for each possible offset in Ω. Notation is provided for time slice $t = 3$.

4.1.1 Variables of HPM-DBNs

To write an HPM as a DBN, we create a chain of discrete state variables like a fHMM for each process in the HPM. Following the notation of Ghahramani and Jordan [1997], we will write $S_t^{(\pi)}$ to mean an indicator vector holding the value of the chain corresponding to process π at time t. Each state variable takes on values from 0 to $d(\pi)$, the duration of the process to which it corresponds, so $S_t^{(\pi)}$ has length $d(\pi)+1$. The dynamics of HPM-DBNs are designed so that a process-chain starts at 0, transitions to 1 when the process begins, and deterministically counts to its duration before transitioning back to 0. In Figure 4.1, these state variables are the square unshaded nodes.

HPM-DBNs also have a layer of input nodes that are not typically present in fHMMs, represented by the shaded triangles in Figure 4.1. These nodes contain information about the timing landmarks in the experiment. Each chain has a corresponding binary input value at every time step called $I_t^{(\pi)}$ that is 1 if a landmark for process π occurred at time t and 0 otherwise. While these nodes may contain information about different kinds of landmarks for the same process (e.g. second stimulus presentation and button press) we assume that the distribution governing the process start time in relation to the landmark time is invariant with respect to the type of landmark and treat all landmarks for a given process the same. The set of nodes influenced by a particular input node depends on the number of offsets allowed for the corresponding process ($\Omega(\pi)$); there is a line from $I_t^{(\pi)}$ to $S_{t+o}^{(\pi)}$ for every value of $o \in \Omega(\pi)$. HPM-DBNs assume that the set of input nodes $\{I_{t-\Omega(\pi)}^{(\pi)}\}$ that influences $S_t^{(\pi)}$ are either all zero or contain only a single one. These constraints reflect the assumption that only a single instance of a process can be occurring at a given time.

Finally, an HPM-DBN contains a continuous observed variable (the shaded circles in Figure 4.1) representing the data at each time step called Y_t. $\{Y\}$ ($\{Y\} = \{Y_t\} \forall t$) does not have its own temporal dynamics, but instead depends on the values of the chains at time t, which we will write $\{S_t\}$, as shorthand for $\{S_t^{(1)}, S_t^{(2)}, \ldots, S_t^{(\Pi)}\}$.

4.1.2 Parameters and Distributions of HPM-DBNs

The set of parameters Ψ of an HPM-DBN contains a vector of parameters called $b^{(\pi)}$ that define a distribution over the state values at $t = 1$ for each chain, transition matrices called $A^{(\pi)}$ defining the temporal dynamics of each chain, weight matrices called $W^{(\pi)}$ that contain the contributions of that chain to the emission distribution over the observed nodes for each chain, and noise variables called σ_v^2 for the emission distribution for each dimension of the data ($v = [1, \ldots, V]$). We now discuss these parameters in more detail,

along with the distributions they define.

First, consider the probability distribution over $S_1^{(\pi)}$ for some chain π, which depends on $I_1^{(\pi)}$, $P(S_1^{(\pi)}|I_1^{(\pi)})$. We define $b^{(\pi)} = \{b_0^{(\pi)}, b_1^{(\pi)}\}$, where the subscript contains the value of $I_1^{(\pi)}$. With these vectors, we can write:

$$\begin{aligned} P(S_1^{(\pi)}|I_1^{(\pi)}) &= \left(\prod_{k=1}^{d(\pi)+1} (b_{0,k}^{(\pi)})^{S_{1,k}^{(\pi)}}\right)^{(1-I_1^{(\pi)})} \left(\prod_{k=1}^{d(\pi)+1} (b_{1,k}^{(\pi)})^{S_{1,k}^{(\pi)}}\right)^{I_1^{(\pi)}} \\ \log P(S_1^{(\pi)}|I_1^{(\pi)}) &= (1 - I_1^{(\pi)})(S_1^{(\pi)})' \log b_0^{(\pi)} + I_1^{(\pi)}(S_1^{(\pi)})' \log b_1^{(\pi)} \end{aligned} \tag{4.1}$$

These equations describe a mixture of multinomial distributions over the values of $S_1^{(\pi)}$, where $I_1^{(\pi)}$ selects which multinomial to use. (Note that it would also be reasonable to fix $S_0^{(\pi)} = 0$ and $I_0^{(\pi)} = 0$ for all π and only deal with the transition matrices $A^{(\pi)}$.)

The probability distribution over $S_t^{(\pi)}$ depends on $S_{t-1}^{(\pi)}$ and some set of the input variables $\{I_{t-\Omega(\pi)}^{(\pi)}\}$, where we assume that if some value of $\{t - \Omega(\pi)\}$ lies outside the legal range of time steps ($t \in [1, T]$) it is removed from the set. This distribution is parameterized by a set of transition matrices $A^{(\pi)} = \{A_\oslash^{(\pi)}, A_o^{(\pi)}\}, \forall o \in \Omega(\pi)$. Each transition matrix in this set covers a different setting of the input variables $I_{t-\Omega(\pi)}^{(\pi)}$ on which $S_t^{(\pi)}$ depends. $A_\oslash^{(\pi)}$ is the transition matrix for the case where all of the $I_{t-\Omega(\pi)}^{(\pi)}$ are 0; that is, none of the landmarks for process π occurred during the legal offset window. $A_o^{(\pi)}$ is the transition

matrix for the case in which $I_{t-o}^{(\pi)} = 1$. The full distribution is then written as:

$$\begin{aligned} P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}) &= \left(\prod_{i=1}^{d(\pi)+1} \prod_{j=1}^{d(\pi)+1} (A_{\oslash,i,j}^{(\pi)})^{S_{t-1,i}^{(\pi)} S_{t,j}^{(\pi)}} \right)^{1-\sum_{o\in\Omega(\pi)} I_{t-o}^{(\pi)}} \times \\ & \prod_{o\in\Omega(\pi)} \left(\left(\prod_{i=1}^{d(\pi)+1} \prod_{j=1}^{d(\pi)+1} (A_{o,i,j}^{(\pi)})^{S_{t-1,i}^{(\pi)} S_{t,j}^{(\pi)}} \right)^{I_{t-o}^{(\pi)}} \right) \\ \log P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}) &= \left(1 - \sum_{o\in\Omega(\pi)} I_{t-o}^{(\pi)} \right) (S_{t-1}^{(\pi)})' \log A_{\oslash}^{(\pi)} S_t^{(\pi)} + \\ & \sum_{o\in\Omega(\pi)} \left(I_{t-o}^{(\pi)} (S_{t-1}^{(\pi)})' \log A_o^{(\pi)} S_t^{(\pi)} \right) \end{aligned} \tag{4.2}$$

As mentioned above, the desired dynamics for each chain are largely known in advance for HPM-DBNs. While we have uncertainty about when the process should begin (i.e. at which time step the chain should go from 0 to 1), once it begins it must count to its duration and return to 0. This means that $A_{\oslash}^{(\pi)}$ is fully deterministic; since it reflects the case where none of the input variables are one, chain π may not transition from 0 to 1 (we do not allow processes to begin spontaneously) and must continue counting if its value is already above 1. (Once a chain begins counting, we ignore the input values until it resets to zero again. This is consistent with our assumption that only one instance of a process is active at a time.) In each transition matrix corresponding to an offset value o, there is only 1 free parameter (e.g. in $A_o^{(\pi)}$, writing $A_o^{(\pi)}(S_{t-1}^{(\pi)}, S_t^{(\pi)})$, the only unknown element is $A_o^{(\pi)}(0,1)$ since $A_o^{(\pi)}(0,0) = 1 - A_o^{(\pi)}(0,1)$). This free parameter closely corresponds to $\Theta_o(\pi)$ in discrete HPMs. In that case, a process was forced to occur following its landmark since $\sum_{o\in\Omega(\pi)} \Theta_o(\pi) = 1$. We can either remove this constraint in HPM-DBNs and let the free parameters of the $A_o^{(\pi)}$ be independent, or we can retain this constraint by renormalizing the free parameters. For instance, for $\Omega(\pi) = [0,1,2]$, $\Theta(\pi) = [0.1, 0.4, 0.5]$, and $I_t^{(\pi)} = 1$, $A_0^{(\pi)}(0,1) = 0.1$, $A_1^{(\pi)}(0,1) = 0.5$, and $A_2^{(\pi)}(0,1) = 1$.

The emission distribution for this model is exactly the one used for HPMs in Chapter

2:

$$
\begin{aligned}
Y_t|\{S_t\} &\sim N(\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)}, \Sigma) \\
P(Y_t|\{S_t\}) &= \frac{1}{(2\pi)^{V/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(Y_t - \sum_{\pi}(W^{(\pi)})'S_t^{(\pi)})'\Sigma^{-1}(Y_t - \sum_{\pi}(W^{(\pi)})'S_t^{(\pi)})) \\
\log P(Y_t|\{S_t\}) &= -\log((2\pi)^{V/2}|\Sigma|^{1/2}) - \frac{1}{2}(Y_t - \sum_{\pi}(W^{(\pi)})'S_t^{(\pi)})'\Sigma^{-1}(Y_t - \sum_{\pi}(W^{(\pi)})'S_t^{(\pi)}) \\
&= -\log((2\pi)^{V/2}|\Sigma|^{1/2}) - \frac{1}{2}(Y_t)'\Sigma^{-1}Y_t + \\
&\quad (Y_t)'\Sigma^{-1}\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)} - \frac{1}{2}(\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)})'\Sigma^{-1}\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)}
\end{aligned}
\tag{4.3}
$$

Here, Y_t is a $V \times 1$ vector, $\{S_t\}$ represents the collection of $|\Pi|$ state variables at time t, $W^{(\pi)}$ is a $(D^{(\pi)}+1) \times V$ matrix containing the response signature parameters for process π augmented with a row of zeros for $S_t^{(\pi)} = 0$, and Σ is a $V \times V$ diagonal covariance matrix containing the σ_v^2 variables.

Finally, let us write down the full log-likelihood of the HPM-DBN model:

$$
\begin{aligned}
P(\{I,S,Y\}|\Psi) &= \prod_{\pi=1}^{\Pi}[P(S_1^{(\pi)}|I_1^{(\pi)}, b^{(\pi)})] \times \\
&\quad \prod_{t=2}^{T}\prod_{\pi=1}^{\Pi}[P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}, \{S_{t-\delta}^{(\pi'\neq\pi)}\}, A^{(\pi)})] \times \\
&\quad \prod_{t=1}^{T} P(Y_t|\{S_t\}, W, \Sigma) \\
\log P(\{I,S,Y\}|\Psi) &= \sum_{\pi=1}^{\Pi}[\log P(S_1^{(\pi)}|I_1^{(\pi)}, b^{(\pi)})] + \\
&\quad \sum_{t=2}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}, \{S_{t-\delta}^{(\pi'\neq\pi)}\}, A^{(\pi)})] + \\
&\quad \sum_{t=1}^{T} \log P(Y_t|\{S_t\}, W, \Sigma)
\end{aligned}
\tag{4.4}
$$

4.2 Modeling Differences and Extensions

HPM-DBNs are not equivalent to the HPMs presented in Chapter 2, nor the extensions presented in Chapter 3. The earlier models based on configurations allowed the explicit inclusion or exclusion of specific arrangements of processes. The removal of configurations from the model entails an arguably more elegant, but technically less expressive distribution over the hidden variables describing the underlying processes.

For instance, the HPMs discussed in earlier chapters allowed multiple overlapping process instances of the same type. HPM-DBNs do not allow this phenomenon. One might think it would be a simple extension, in that the state of a chain could be the sum of the values for each process instance. The difficulty is in disambiguating the contribution of each instance. For example, if the state value of a chain at a particular time is 7, there are several possible interpretations. This node might represent a single process that began 7 time steps ago, or it might represent two instances that began 2 and 5 time steps ago, or two instances that began 3 and 4 time steps ago, or three instances that began 1 and 2 and 4 time steps ago, etc.

Another difference between configuration-based HPMs and HPM-DBNs is that configurations easily accomodate process ordering constraints, whereas HPM-DBNs as discussed so far do not. By choosing the set of configurations carefully, constraints on process ordering are encoded by simply including configurations that respect the constraint and excluding those that do not. In HPM-DBNs as discussed so far, this type of constraint is not present. The start times of the process chains are treated independently, and all possibilities are considered.

Process ordering constraints can be reintroduced into HPM-DBNs by adding arrows from one chain to another, connecting time steps t of the first chain to time steps $t + \delta$ of another, where δ is the required temporal delay between the processes. An example of this is shown in Figure 4.2, where in this case we want the Decide process to begin before the PressButton process. We do not need any new parameters to achieve this constraint; we simply change the expression for $P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\})$ slightly to also depend on $S_{t-\delta}^{(\pi')}$,

where π' is the chain on which π depends:

$$\begin{aligned} P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, S_{t-\delta}^{(\pi')}, \{I_{t-\Omega(\pi)}^{(\pi)}\}) &= \left(\prod_{i=1}^{d(\pi)+1}\prod_{j=1}^{d(\pi)+1}(A_{\oslash,i,j}^{(\pi)})^{S_{t-1,i}^{(\pi)}S_{t,j}^{(\pi)}}\right)^{1-\delta(S_{t-\delta}^{(\pi')}>0)\sum_{o\in\Omega(\pi)}I_{t-o}^{(\pi)}} \times \\ & \prod_o\left(\left(\prod_{i=1}^{d(\pi)+1}\prod_{j=1}^{d(\pi)+1}(A_{o,i,j}^{(\pi)})^{S_{t-1,i}^{(\pi)}S_{t,j}^{(\pi)}}\right)^{\delta(S_{t-\delta}^{(\pi')}>0)I_{t-o}^{(\pi)}}\right) \\ \log P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, S_{t-\delta}^{(\pi')}, \{I_{t-\Omega(\pi)}^{(\pi)}\}) &= \left(1-\delta(S_{t-\delta}^{(\pi')}>0)\sum_{o\in\Omega(\pi)}I_{t-o}^{(\pi)}\right)(S_{t-1}^{(\pi)})'\log A_{\oslash}^{(\pi)}S_t^{(\pi)} + \\ & \delta(S_{t-\delta}^{(\pi')}>0)\sum_{o\in\Omega(\pi)}\left(I_{t-o}^{(\pi)}(S_{t-1}^{(\pi)})'\log A_o^{(\pi)}S_t^{(\pi)}\right) \end{aligned} \tag{4.5}$$

where the delta function $\delta(\cdot)$ returns 1 if its argument is true and 0 otherwise. This equation simply defines the conditions under which each of the transition matrices is to be used. If $\delta(S_{t-\delta}^{(\pi')} > 0)I_{t-o}^{(\pi)}$ is 1, then it is true that π' has begun and $I_{t-o}^{(\pi)}$ is 1, so the appropriate transition matrix is $A_o^{(\pi)}$. If $1-\delta(S_{t-\delta}^{(\pi')} > 0)\sum_o I_{t-o}^{(\pi)}$ is 1, then it is true that either π' has not yet started (its value is still 0) and/or none of the input values in the legal window $t-\Omega(\pi)$ are 1. In this case, the conditions for the start of process π are not met, and we use $A_{\oslash}^{(\pi)}$.

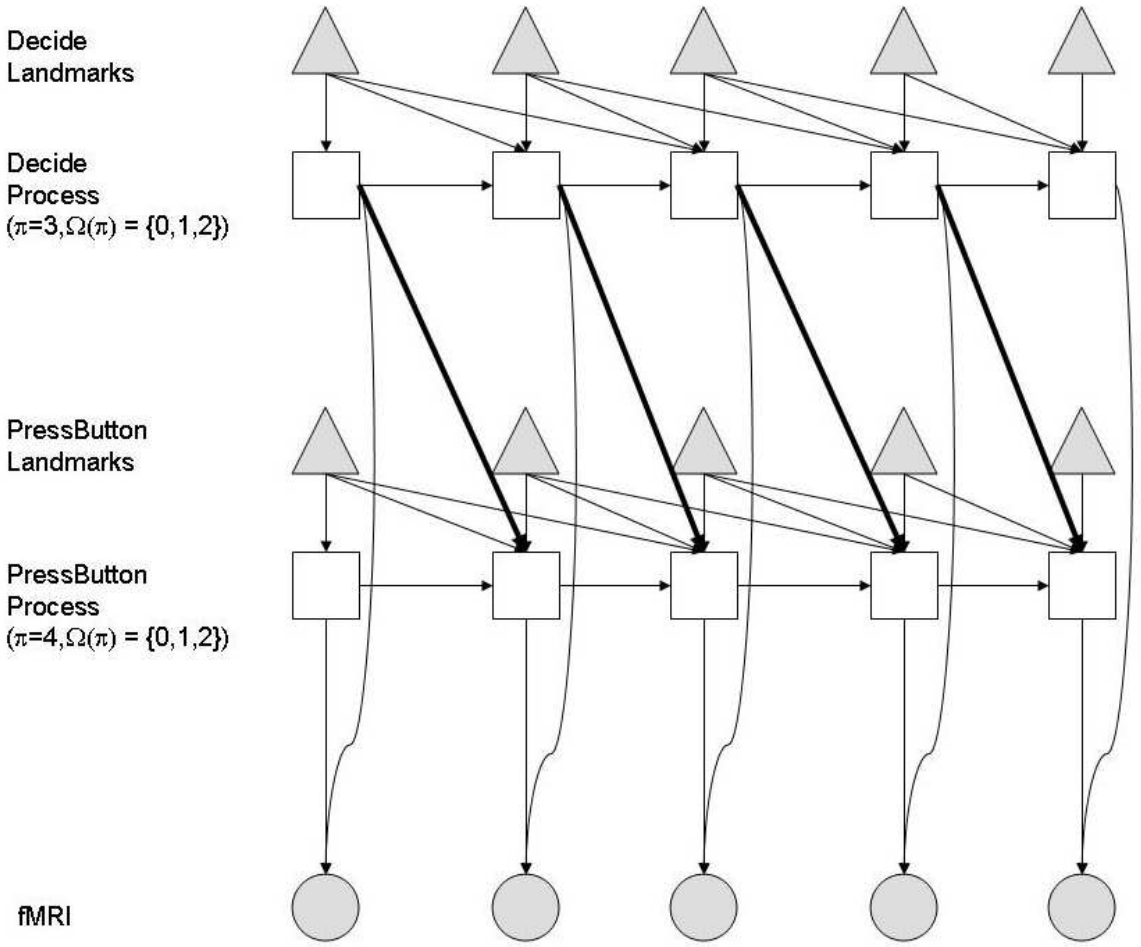

Figure 4.2: Example process ordering constraint. The bold arrows encode the dependece of the PressButton chain on the Decide chain.

In designing a model with process ordering constraints, one must be aware of the possibility that a ill-designed constraints could potentially disallow a process instance. For instance, if the start time windows for two processes are identical, and we constrain the model such that one must begin strictly before the other, then choosing the last possible start time for the first process precludes the second process from occuring at all, since all of its possible start times violate the ordering constraint. In some cases this consequence may be acceptable; in other cases it may be undesirable. We will leave this concern to the model designer.

4.3 Algorithms for HPM-DBNs

HPM-DBNs without process ordering constraints can be treated just like fHMMs for doing inference and learning, since the addition of an observed input layer and restricting the transition dynamics does not add any complexity to the model. Appendix C discusses these algorithms. Below, we discuss algorithms for HPM-DBNs with process ordering constraints. Introducing dependencies between process chains in the model to encode these constraints creates couplings between the chains beyond just the shared responsibility for the output variable. As a result, we can no longer use standard fHMM algorithms to do inference and learning in this model. Appendix E develops a variational approximation for HPM-DBNs with process ordering constraints based on the structured approximation for fHMMs. The rest of this chapter discusses Markov Chain Monte Carlo (MCMC) sampling algorithms (MacKay [1998, 2003]) to do Bayesian inference and learning in HPM-DBNs. Note that the Bayesian inference approach is not specific to the HPM-DBN representation; we could train HPM-DBNs using maximum likelihood techniques (with or without regularization), and/or we could develop Bayesian inference techniques for the HPMs of Chapter 2 as well.

4.4 A MCMC Sampling Algorithm for HPM-DBNs

The inference problem in HPM-DBNs is to recover the sequence of hidden states $\{S^{(\pi)}\}$ for each chain π given the observed inputs $\{I\}$, the observed outputs $\{Y\}$, and the parameters of the model Ψ. The learning problem is to estimate the parameters of the model Ψ given the observed inputs $\{I\}$ and the observed outputs $\{Y\}$.

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method that addresses both inference and learning (MacKay [2003, 1998]). MCMC methods iteratively generate sam-

ples of a hidden variable x. The sample at iteration t called $\{x^{(t)}\}$ depends on the sample at iteration $t-1$ ($\{x^{(t-1)}\}$), and the sequence of iterations defines a Markov chain. As $t \rightarrow \infty$, $\{x^{(t)}\}$ tends to the true distribution $P(x)$. In Gibbs sampling, each dimension x_i of the hidden variable is sampled from its full conditional distribution given all other dimensions.

To do Gibbs sampling for learning in HPM-DBNs, the hidden variables to be sampled are $\{S\}$ and Ψ. We can think of the dimensions of these hidden variables as a grouping of the components that can be sampled individually at each iteration, conditioned on everything else. In our case, we sample the values of each chain in turn ($\{S^{(\pi)}\}$ one π at a time), and then sample the values of the parameters of the model:

1. Initialize values for $\{S\}^{(0)}$ and $\Psi^{(0)}$.
2. For $iter = 1 : N$
   (a) For $\pi = 1 : \Pi$:
      i. Sample $\{S^{(\pi)}\}^{(iter)}$ from $P(\{S^{(\pi)}\}|\{I\}, \{S^{(\pi' \neq \pi)}\}^{(iter-1)}, \{Y\}, \Psi^{(iter-1)})$.
   (b) Sample $\Psi^{(iter)}$ from $P(\Psi|\{I\}, \{S^{(\pi)}\}^{(iter)}, \{Y\})$.

If the parameters are known, we can do inference by just sampling the chains and not updating the parameters (i.e. omitting step 2b).

Below, we examine each step in greater detail. Section 4.4.1 adapts a sampling method presented in Scott [2002] for Hidden Markov Models (HMMs) similar to the forward-backward algorithm to sample each chain while holding the other chains and parameters fixed. Section 4.4.2 gives the full conditional distributions for each parameter to be sampled.

4.4.1 Sampling Chains in HPM-DBNs

In Step 2(a)i above, we wish to sample $\{S^{(\pi)}\}$ from its full conditional distribution, given all other variables in the model: $P(\{S^{(\pi)}\}|\{I\}, \{S^{(\pi' \neq \pi)}\}^{(iter-1)}, \{Y\}, \Psi^{(iter-1)})$. We can accomplish this with a simple modification to an existing algorithm for sampling the hidden states of a Hidden Markov Model (HMM) (Scott [2002]).

The method described in Scott [2002] is based on the forward-backward algorithm for HMMs (Rabiner [1989]). The forward-backward algorithm efficiently computes the probability distribution over the hidden variables in an HMM (the unobserved state sequence) given the observed data and parameters. Let us call the hidden states of the HMM S_t and

the data Y_t, for time steps $t = [1, \ldots, T]$, and let the set of all parameters of the HMM be Ψ. The forward pass computes matrices $\mathbf{F_t}$ for $t = [2, \ldots, T]$ where $\mathbf{F_t}(r, s) \propto P(S_{t-1} = r, S_t = s, Y_t | Y_{1:t-1}, \Psi)$, where $Y_{1:t}$ denotes the sequence $\{Y_\tau\}$ where $\tau = [1, \ldots, t]$. The backward pass computes matrices $\mathbf{B_t}$ where $\mathbf{B_t}(r, s) \propto P(S_{t-1} = r, S_t = s | Y_{1:T}, \Psi)$. Proportionality for each matrix is reconciled by $\sum_r \sum_s \mathbf{F_t}(r, s) = 1$ and $\sum_r \sum_s \mathbf{B_t}(r, s) = 1$. Note that this normalization means that $\mathbf{F_t}(r, s) = P(S_{t-1} = r, S_t = s | Y_{1:t}, \Psi)$.

A method for recursively sampling the states of an HMM on the backward pass is presented in Scott [2002]. The stochastic backward recursion begins by sampling a value for S_T from the last result of the forward pass, summed over all possible state values for S_{T-1}. That is, S_T is sampled from $\sum_r \mathbf{F_T}(r, :) = \sum_r P(S_T, S_{T-1} = r | Y_{1:T}, \Psi)$. Given a sampled value $\tilde{S}_t$, S_{t-1} is recursively sampled from the appropriate column of $\mathbf{F_t}$. That is, we draw S_{t-1} from $\mathbf{F_t}(:, \tilde{S}_t)$.

The extension to HPM-DBNs is straightforward. We consider three cases: those that influence another chain, those that are influenced by another chain, and those that are only coupled to other chains through the output variables.

Case 1: sampling a chain without direct links to other chains

For a chain without direct links to other chains, which we will call chain $\pi = 1$, the forward pass computes $P(Y_{1:t}, S_t^{(1)} | \Psi, I^{(1)})$ recursively for all t, and the backward pass samples $S_t^{(1)}$ from $P(S_t^{(1)} | Y_{1:T}, \Psi)$ recursively for all t. More specifically, the forward recursion constructs $(d(1) + 1) \times (d(1) + 1)$ matrices $\mathbf{F_t} \propto P(S_{t-1}^{(1)}, S_t^{(1)}, Y_t | Y_{1:t-1}, \Psi)$ for $t = [2, \ldots, T]$ as follows:

$$
\begin{aligned}
\mathbf{F_2}(r, s) &\propto P(S_1^{(1)} = r | \{I^{(1)}\}, \Psi) P(Y_1 | S_1^{(1)} = r, \{S_1^{(\pi \neq 1)}\}, \Psi) \times \\
&\quad P(S_2^{(1)} = s | S_1^{(1)} = r, \{I^{(1)}\}, \Psi) P(Y_2 | S_2^{(1)} = s, \{S_2^{(\pi \neq 1)}\}, \Psi) \\
\mathbf{F_t}(r, s) &\propto P(S_{t-1}^{(1)} = r | Y_{1:t-1}, \{I^{(1)}\}, \Psi) P(S_t^{(1)} = s | S_{t-1}^{(1)} = r, \{I^{(1)}\}, \Psi) \\
&\quad P(Y_t | S_t^{(1)} = s, \{S_t^{(\pi \neq 1)}\}, \Psi) \\
&= \sum_q [\mathbf{F_{t-1}}(q, r)] P(S_t^{(1)} = s | S_{t-1}^{(1)} = r, \{I^{(1)}\}, A^{(1)}) \\
&\quad P(Y_t | S_t^{(1)} = s, \{S_t^{(\pi \neq 1)}\}, W, \Sigma)
\end{aligned}
\tag{4.6}
$$

where proportionality is reconciled with $\sum_r \sum_s \mathbf{F_t}(r, s) = 1$. All of these distributions are defined in the probabilistic model described in Section 4.1. The stochastic backward

pass samples a value from the distribution specified by the appropriate column of the $\mathbf{F_t}$ matrix as follows (the $\tilde{\cdot}$ indicates a sampled value):

$$\begin{aligned} \tilde{S_T^{(1)}} &\sim \sum_r \mathbf{F_T}(r, \cdot) \\ \tilde{S_{t-1}^{(1)}} &\sim \mathbf{F_t}(\cdot, \tilde{S_t^{(1)}}) \end{aligned} \tag{4.7}$$

Case 2: sampling a chain with outgoing links to another chain

Now consider chain $\pi = 2$ with links from its nodes at times t to the nodes of a different chain (let us call this chain $\pi = 3$) at $t + \delta$. Since Gibbs sampling assumes the values of all variables not being sampled in a given step are fixed and known, we can treat the fixed, known values $\{S^{(3)}\}$ essentially as additional observations in the model, as we did with $\{I\}$ and $\{Y\}$ in Case 1. Now the forward recursion becomes:

$$\begin{aligned} \mathbf{F_t}(r, s) \quad \propto \quad & \sum_q [\mathbf{F_{t-1}}(q, r)] P(S_t^{(2)} = s | S_{t-1}^{(2)} = r, \{I^{(2)}\}, A^{(2)}) \\ & P(Y_t | S_t^{(2)} = s, \{S_t^{(\pi \neq 1)}\}, W, \Sigma) P(S_{t+\delta}^{(3)} | S_{t+\delta-1}^{(3)}, S_t^{(2)}, \{I^{(3)}\}, A^{(3)}) \end{aligned} \tag{4.8}$$

with appropriate adjustments for end effects, and with proportionality again reconciled with $\sum_r \sum_s \mathbf{F_t}(r, s) = 1$. The backward sampling is the same once the $\mathbf{F_t}$ are constructed.

Case 3: sampling a chain with incoming links from another chain

Chain $\pi = 3$ described above has incoming links from chain $\pi = 2$. This again requires a slight change in the construction of the $\mathbf{F_t}$, but again the change is simple since we can treat the values of $\{S^{(2)}\}$ as fully observed when sampling $\{S^{(3)}\}$. Here the forward matrices are constructed as follows:

$$\begin{aligned} \mathbf{F_t}(r, s) \quad \propto \quad & \sum_q [\mathbf{F_{t-1}}(q, r)] P(S_t^{(3)} = s | S_{t-1}^{(3)} = r, S_{t-\delta}^{(2)} \{I^{(3)}\}, A^{(3)}) \\ & P(Y_t | S_t^{(3)} = s, \{S_t^{(\pi \neq 1)}\}, W, \Sigma) \end{aligned} \tag{4.9}$$

Once more, we must normalize such that $\sum_r \sum_s \mathbf{F_t}(r, s) = 1$ and adjust for end effects, after which the backward recursion is the same.

4.4.2 Updating the HPM-DBN Parameters

Having sampled values for the hidden states $\{S\}$ on a given iteration of the Gibbs sampling procedure, the next step is to sample new values for the parameters Ψ. Recall that the distribution to be sampled from is the full conditional distribution of Ψ given everything else in the model:

$$P(\Psi|\{I,S,Y\}) \quad \propto \quad P(\{I,S,Y\}|\Psi)P(\Psi) \tag{4.10}$$

The likelihood $P(\{I,S,Y\}|\Psi)$ is defined in Section 4.1. Let us define the priors over the components of Ψ to be a priori independent:

$$\begin{aligned}
P(\Psi) &= P(\Sigma)\prod_{\pi=1}^{\Pi} P(b^{(\pi)})P(A^{(\pi)})P(W^{(\pi)}) \\
&= P(\Sigma)\prod_{\pi=1}^{\Pi} P(b_0^{(\pi)})P(b_1^{(\pi)})P(W^{(\pi)})P(A_{\oslash}^{(\pi)})\prod_{o\in\Omega(\pi)} P(A_o^{(\pi)}) \\
\log P(\Psi) &= \log P(\Sigma) + \sum_{\pi=1}^{\Pi}[\log P(b_0^{(\pi)}) + \log P(b_1^{(\pi)}) + \log P(W^{(\pi)}) + \\
&\quad \log P(A_{\oslash}^{(\pi)}) + \sum_{o\in\Omega(\pi)} \log P(A_o^{(\pi)})]
\end{aligned} \tag{4.11}$$

So for some normalizing constant c we have:

$$
\begin{aligned}
\log P(\Psi|\{I,S,Y\}) \;=\; & c + \sum_{\pi=1}^{\Pi}[\log P(S_1^{(\pi)}|I_1^{(\pi)}, b^{(\pi)})] + \\
& \sum_{t=2}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}, \{S_{t-\delta}^{(\pi'\neq\pi)}\}, A^{(\pi)})] + \\
& \sum_{t=1}^{T}\log P(Y_t|\{S_t\}, W, \Sigma) + \\
& \log P(\Sigma) + \sum_{\pi=1}^{\Pi}[\log P(b_0^{(\pi)}) + \log P(b_1^{(\pi)}) + \log P(W^{(\pi)}) + \\
& \log P(A_{\oslash}^{(\pi)}) + \sum_{o\in\Omega(\pi)}\log P(A_o^{(\pi)})]
\end{aligned}
\tag{4.12}
$$

For each element of Ψ in turn, we can choose an appropriate prior distribution and derive its full conditional distribution by grouping terms not involving the parameter of interest into the normalizing constant c. We summarize these distributions below; Appendix D provides more detail on how each distribution was derived.

First consider the parameters $b^{(\pi)}$. Since $b_0^{(\pi)}$ contains the parameters of a multinomial distribution, we use the conjugate prior, a Dirichlet distribution, with parameters $\alpha_0^{(\pi)}$ (a $D^{(\pi)} \times 1$ vector). Then the full conditional distribution from which we sample a new value of $b_0^{(\pi)}$ at each Gibbs iteration is also Dirichlet, with parameters $(1 - I_1^{(\pi)})S_1^{(\pi)} + \alpha_0^{(\pi)}$. The parameters $b_1^{(\pi)}$ are treated similarly.

Next consider the transition matrices. Recall that each $A_o^{(\pi)}$ contains only one free parameter, which we will call $\rho_o^{(\pi)}$. Since $\rho_o^{(\pi)}$ is effectively the parameter of a Binomial distribution, the conjugate prior $P(\rho_o^{(\pi)})$ is a Beta distribution with parameters $\beta_o^{(\pi)}$ and $\gamma_o^{(\pi)}$. This results in another Beta distribution to sample at each Gibbs iteration, with parameters

$$\sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')} > 0)I_{t-\omega}^{(\pi)}S_{t-1}^{(\pi),0}S_t^{(\pi),1}] + \beta_\omega^{(\pi)}$$

and

$$\sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')} > 0)I_{t-\omega}^{(\pi)}S_{t-1}^{(\pi),0}S_t^{(\pi),0}] + \gamma_\omega^{(\pi)}$$

The weight matrices $W^{(\pi)}$ containing the contributions of each chain to the output are $(d(\pi)+1) \times V$ and we will sample them one row ($W_d^{(\pi)}$) at a time. Under a multivariate Gaussian prior with parameters $\mu_d^{(\pi)}$ (a $V \times 1$ vector) and $\Sigma_d^{(\pi)}$ (a $V \times V$ matrix), we obtain a multivariate Gaussian sampling distribution with parameters

$$\left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)}]\Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right)^{-1}\left(\Sigma^{-1}\sum_{t=1}^{T}[S_{t,d}^{(\pi)}(Y_t - \frac{1}{2}\sum_{\pi' \neq \pi}(W^{(\pi')})'S_t^{(\pi')})] + (\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}\right)$$

and

$$\left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)}]\Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right)^{-1}$$

Finally, we can treat the elements of the diagonal of Σ independently, so we define an inverse-Gamma prior for each σ_v^2 with parameters η_v and ν_v. This results in an inverse-Gamma to sample from, with parameters $\frac{1}{2}T + \eta_v$ and $\nu_v + \frac{1}{2}\sum_{t=1}^{T}[(Y_{t,v} - \mu_{t,v})^2]$.

In summary, this chapter has presented HPM-DBNs, a Dynamic Bayes Net similar to a factorial HMM. The assumptions made by HPM-DBNs are similar to those made by the HPMs presented in the previous chapters, but as we have discussed, they are not the same. In particular, the HPM-DBNs we described cannot accomodate temporally overlapping instances of the same process, whereas discrete HPMs can. Discrete HPMs generally have more flexibility than HPM-DBNs in restricting the hypothesis space through configurations. Another example of this flexibility is that discrete HPMs can use different sets of configurations for different trials, whereas the HPM-DBN we described maintains the same structure for all trials. For experiment designs and models where this flexibility is not necessary, the major difference between discrete HPMs and HPM-DBNs is the use of EM versus sampling to learn the model parameters. Based on the lack of convergence for the sampling algorithm for continuous HPMs in such a sparse, high-dimensional setting, we anticipated similar problems and did not implement the sampling algorithm for HPM-DBNs. The next chapter presents experimental results for the HPMs discussed in Chapters 2 and 3.

Chapter 5

Experimental Results

In this chapter, we present experimental results using HPMs. We first demonstrate the performance of HPMs on a synthetic dataset for which we know our modeling assumptions are sound and about which we have ground truth. We then give an analysis of the real sentence-picture fMRI dataset exploring the extensions described in Chapter 3.

5.1 Synthetic Data

In this section, we explore the performance of HPMs on synthetic data. The benefit of using synthetic data is the ability to compare results to ground truth. With synthetic data, we can determine how well the learned process response signature parameters match the true responses that generated the data (something for which we have no ground truth in the real data). We can also justify the use of cross-validated data log-likelihood as an evaluation metric by showing that it selects the model with the correct number of processes on synthetic data.

Of course, the drawback of synthetic data is that we cannot really know whether it is a good approximation to real data. In fact, since we generate synthetic data from a true HPM and add independent, identically distributed, Gaussian noise to each point in space and time, we expect that the synthetic data is significantly easier to model than real data, which may or may not be consistent with the HPM assumptions and certainly has more complex noise properties. For example, all of the synthetic data experiments were run with the standard version of HPMs, and as detailed below, the standard version was successful. However, as described later on, standard HPMs suffered from overfitting in the real data experiments, and regularization and basis functions were needed to overcome the problem.

5.1.1 Data Generation

The synthetic data was created to roughly imitate the real sentence-picture experiment. We created datasets using two synthetic processes (ViewPicture and ReadSentence), three processes (adding Decide), and four processes (adding PressButton). For each experiment, all of the voxels responded to all of the processes. It is unlikely that all the voxels in the brain would respond to all processes in an experiment, but we wished to test HPMs in the most challenging setting possible: maximal spatial-temporal overap among the HRFs of different processes.

For each dataset, the HRF for each process was generated by convolving a boxcar function indicating the presence of the stimulus with a gamma function (following Boynton et al. [1996]) with parameters $\{a, \tau, n\}$ of the form

$$h(t) = a * \frac{(\frac{t}{\tau})^{n-1} exp(\frac{-t}{\tau})}{\tau(n-1)!}.$$

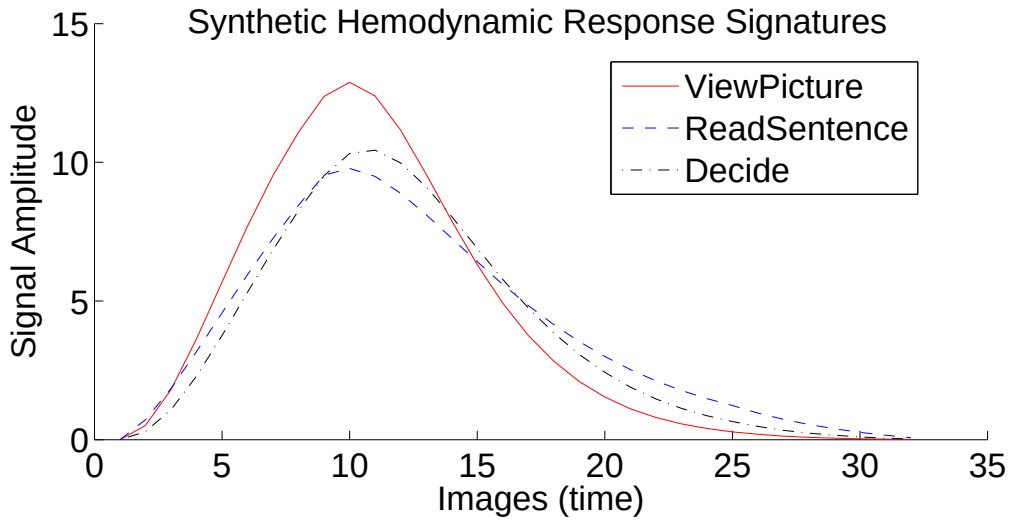

Figure 5.1: Noise-free hemodynamic response functions of the processes in the synthetic data.

The parameters for the gamma functions were $\{8.22, 1.08, 3\}$ for ViewPicture, $\{8, 2.1, 2\}$ for ReadSentence, and $\{7.5, 1.3, 3\}$ for Decide. The processes' gamma functions were

convolved with a 4-second boxcar, matching the experiment timeline. These responses are shown in Figure 5.1.

The ViewPicture and ReadSentence processes had offset values of $\{0, 1\}$, meaning their onsets could be delayed 0 or 1 images (0 or 0.5 seconds) from their corresponding stimuli. The Decide process had offset values of $\{0, 1, 2, 3, 4, 5\}$, meaning its onset could be delayed 0-5 images (0-2.5 seconds) from its corresopnding stimulus, which in each case was the second stimulus presentation, whether it was a picture or a sentence. The synthetic HPMs were roughly equivalent to models used in the real data experiments (HPM-2-U, HPM-3-U, and HPM-4-U, described below).

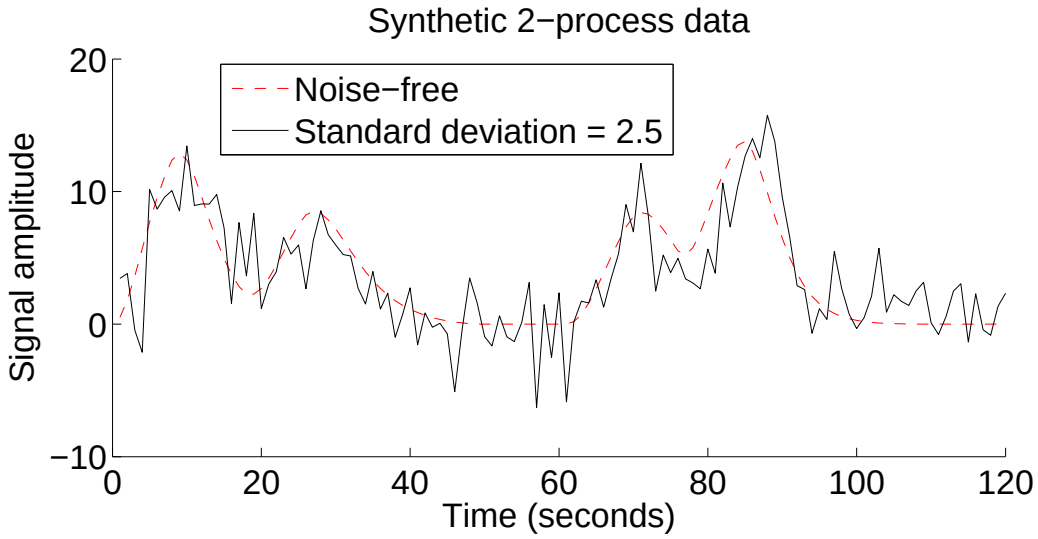

Figure 5.2: Two-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial is the reverse.

To generate the data for a new trial in the two-process dataset, we first chose which stimulus would come first (picture or sentence). We required that there be an equal number of picture-first and sentence-first trials. When a picture stimulus occurred, we selected an offset from the ViewPicture offsets according to the distribution specified by for that process and added it to the stimulus onset time to get the process start time. Then we added the ViewPicture HRF (over all voxels) to the synthetic data beginning at that start time. Sentences were dealt with similarly, where overlapping HRFs summed linearly. Finally, we added Gaussian noise with mean 0 and standard deviation 2.5 independently to every voxel at every time point. The three-process dataset was generated similarly, except that

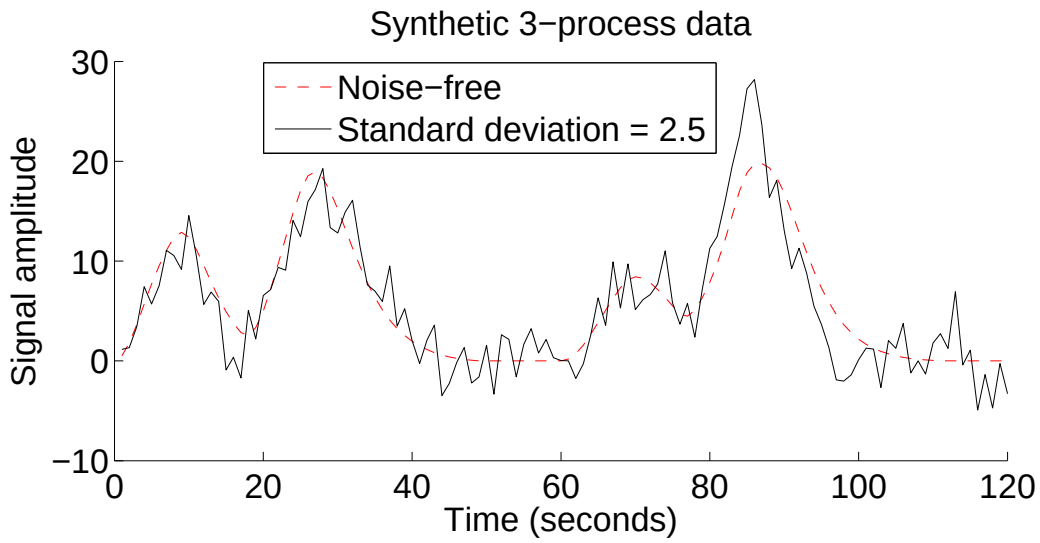

Figure 5.3: Three-process synthetic data: a single voxel timecourse for two trials, with and without noise. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, which starts at time 60, is the reverse. The second peak in each trial is higher than in Figure 5.2 because the three-process data includes activation for the Decide process.

each trial included a Decide process as well, whose onset was chosen according to the timing distribution of that process, and added to the second stimulus onset time to get the process start time. Figure 5.2 shows the timecourse of a single voxel for a sequence of two trials from the two-process dataset (with and without noise), and Figure 5.3 shows the same for the three-process dataset. In both cases, the first trial is a picture followed by a sentence; the second trial is the reverse. The same general process was used to generate four-process data.

5.1.2 Estimation of the Hemodynamic Responses

To test whether HPMs can accurately recover the true process response signatures underlying the data, we compared the learned response signatures to the true responses in the two and three-process synthetic datasets. These datasets each had only 2 voxels (the small number of voxels was chosen to easily show the learned responses, even though the learning problem is easier with more informative voxels). In each case, we trained the model on 40 trials (the number of trials we have in the real data) using standard HPMs (no regularization or basis functions).

In both datasets, the identity of the process instances in each trial were provided to the learning algorithm via the configurations, but their timings were not. This corresponds to reasonable assumptions about the real fMRI data. Since we know the sequence of the stimuli, it is reasonable to assume that the process instances match that sequence. However, we do not know the delay between the stimulus presentation and the beginning of the cognitive process(es) associated with it. This is consistent with the sets of configurations used in the real data experiments.

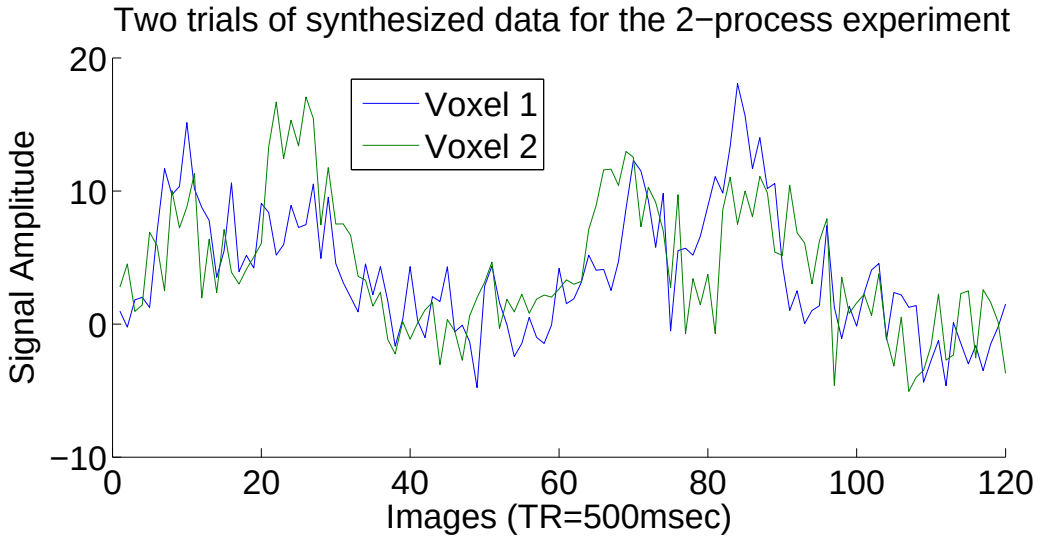

Figure 5.4: A sequence of two trials of synthetic data for the 2-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, starting at time 60, is the reverse.

A sequence of two trials of the two-process data are shown in Figure 5.4, and the learned responses for each process in each voxel are shown in Figure 5.5. Note that the

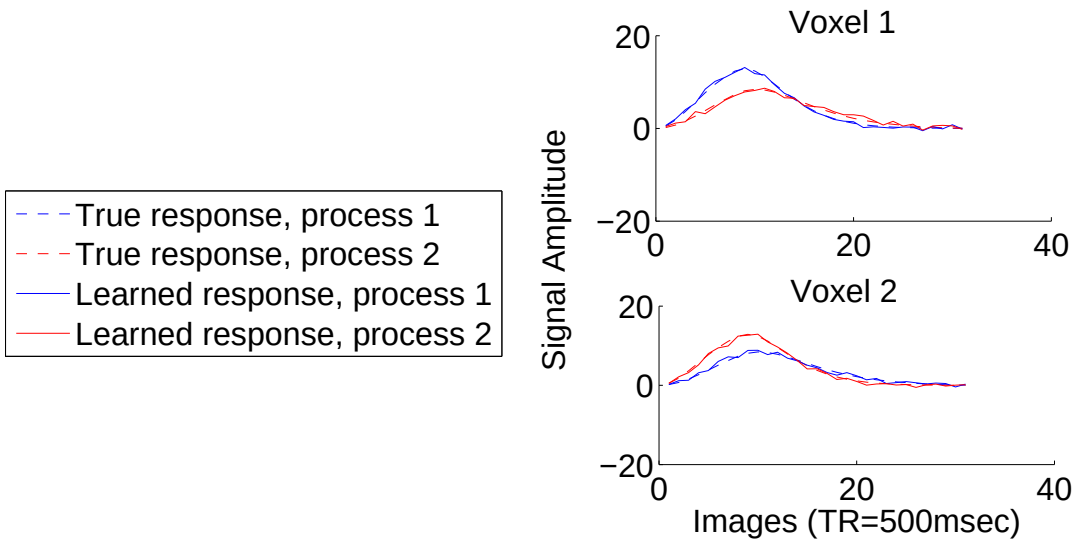

Figure 5.5: Comparison of the learned vs. true process response signatures in the two-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.2647.

learned responses are reasonably smooth, even though this assumption was not provided to the learner. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.2647, and the estimated standard deviations for the voxels are 2.4182 and 2.4686 (compare with the true value of the standard deviation of the noise added to the training data, which was 2.5). Figures 5.6 and 5.7 are the corresponding plots for the three-process data. In this case, the mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.4427, and the estimated standard deviations for the voxels are 2.4316 and 2.4226. The parameters of the timing distributions for the processes were also estimated accurately in both the two-process and three-process experiments. The mean squared error averaged over the 10 θ parameters in the three-process experiment was 0.01.

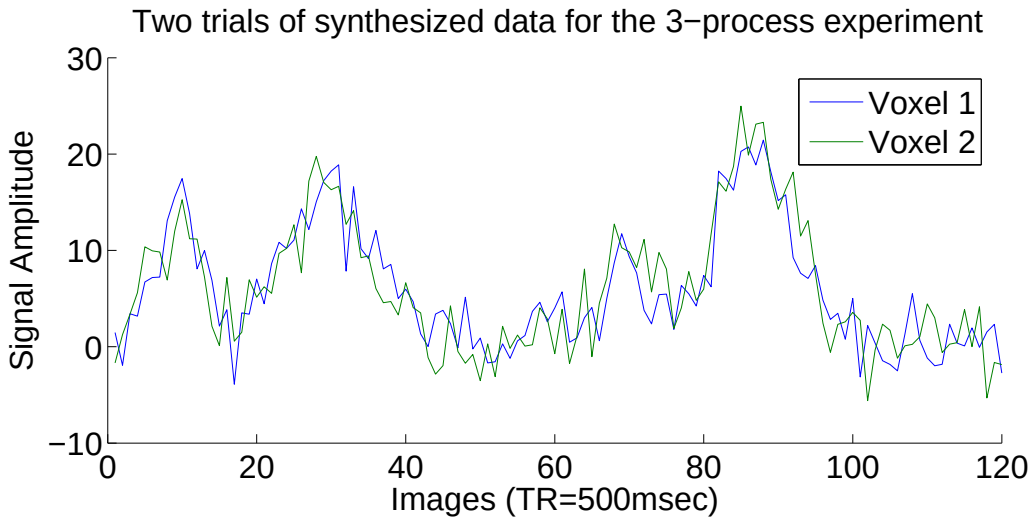

Figure 5.6: A sequence of two trials of synthetic data for the 3-process experiment using 2 voxels. The first trial (the left half of the time series) is a picture followed by a sentence; the second trial, starting at time 60, is the reverse.

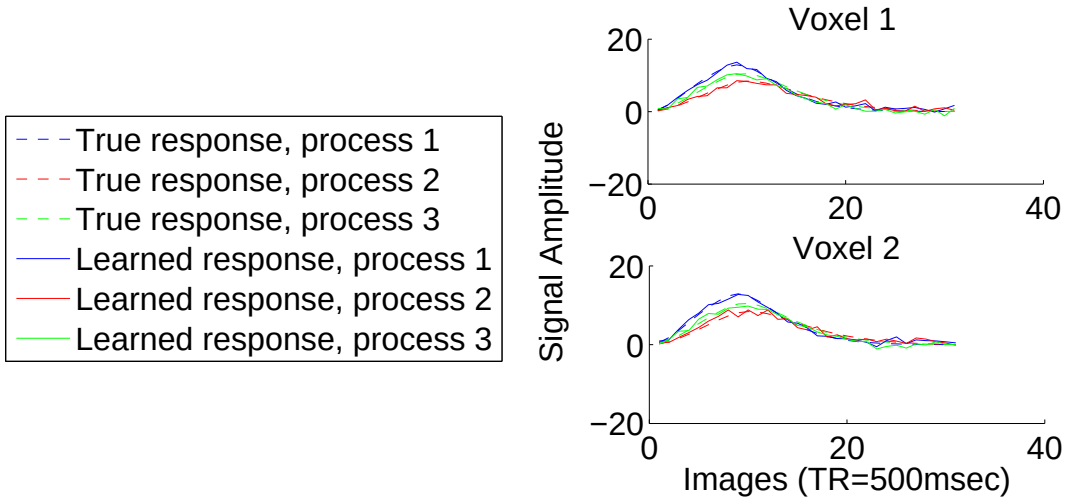

Figure 5.7: Comparison of the learned vs. true process response signatures in the synthetic three-process data for two voxels. The mean squared error between the learned and true responses averaged over timepoints, processes, and voxels is 0.4427.

5.1.3 Choosing the Number of Processes to Model

Another interesting question to approach with synthetic data is whether the use of cross-validated data log-likelihood for model selection in HPMs is justified. One might be concerned about bias toward HPMs with larger numbers of processes, but we expect that the potential for overfitting will be mitigated by the use of cross-validation. In fact, there can be a slight bias, even when using separate test data, in favor of models that allow a larger number of possible configurations. This bias occurs because we compute the posterior distribution over configurations for each test trial, meaning that we can adjust the mixture of configurations to best explain each test trial. Since we can make this adjustment at test time, there can be a bias toward more flexible models, like those with a larger number of possible configurations. There can also be a bias during training toward configurations that cover longer intervals of time within the trial. We attempt to control this source of bias by filling in the mean training trial for any time points that are not influenced by the process instances when we compute the data log-likelihood. As we show below, the impact of these potential biases in our synthetic data experiments does not prevent HPMs from recovering the correct model.

To investigate this question, we generated training sets of 40 examples each using 2, 3, and 4 processes in the same fashion as the previous experiments. For each training set, we trained HPMs with 2, 3, and 4 processes each. Additionally, we generated test sets of 100 examples each using 2, 3, and 4 processes. Every training and test set had 100 voxels. For each test set, we used each of the HPMs trained on its corresponding training set to evaluate the log-likelihood of the test data under the model. More specifically, we used the log-likelihood of the test data under the configuration with the highest posterior probability for each trial. In each case, the model with the best score was the one with the correct number of processes. We performed this experiment with independently generated training and test sets 30 times with consistent results. The average test-set log-likelihoods are shown in Table 5.1, along with further repetitions of this experiment with decreasing training set sizes. As expected, we see increased variability with smaller training sets, but even for small numbers of examples, the HPM with the highest test data log-likelihood has the same number of processes as were used to generate the data.

5.2 Sentence-Picture Verification fMRI Data

Here we present results on the sentence-picture verification fMRI dataset described above. Recall that in this dataset (Keller et al. [2001]), participants were presented with a sequence of 40 trials. In half of the trials participants were shown a picture (involving vertical

Number of Training Trials	Number of Processes in HPM	2 Process Data	3 Process Data	4 Process Data
40	2	$\mathbf{-5.64 \pm 0.00444}$	-7.93 ± 0.0779	-7.72 ± 0.0715
40	3	-7.47 ± 0.183	$\mathbf{-5.66 \pm 0.00391}$	-5.72 ± 0.00504
40	4	-7.19 ± 0.0776	-5.687 ± 0.00482	$\mathbf{-5.65 \pm 0.00381}$
20	2	$\mathbf{-2.87 \pm 0.204}$	-3.80 ± 0.192	-3.70 ± 0.606
20	3	-4.00 ± 0.0461	$\mathbf{-2.86 \pm 0.00597}$	-2.87 ± 0.00276
20	4	-3.91 ± 0.0319	-2.89 ± 0.00320	$\mathbf{-2.85 \pm 0.00364}$
10	2	$\mathbf{-1.44 \pm 0.245}$	-2.07 ± 0.0653	-1.96 ± 0.0665
10	3	-1.99 ± 0.119	$\mathbf{-1.47 \pm 0.0231}$	-1.47 ± 0.00654
10	4	-1.95 ± 0.0872	-1.49 ± 0.0195	$\mathbf{-1.46 \pm 0.00427}$
6	2	$\mathbf{-2.87 \pm 0.204}$	-1.32 ± 0.0363	-1.34 ± 0.297
6	3	-4.00 ± 0.0461	$\mathbf{-0.923 \pm 0.0130}$	-0.928 ± 0.0126
6	4	-3.91 ± 0.0319	-0.933 ± 0.0149	$\mathbf{-0.921 \pm 0.00976}$
2	2	$\mathbf{-3.75 \pm 0.00710}$	-7.23 ± 0.0456	-4.62 ± 0.0383
2	3	-5.36 ± 0.0689	$\mathbf{-6.99 \pm 0.0823}$	-3.96 ± 0.0252
2	4	-5.08 ± 0.0704	-7.02 ± 0.0853	$\mathbf{-3.91 \pm 0.0241}$

Table 5.1: For each training set, the table shows the average (over 30 runs) test set log-likelihood of each of 3 HPMs (with 2, 3, and 4 processes) on each of 3 synthetic data sets (generated with 2, 3, and 4 processes). Each cell is reported as mean $\pm$ standard deviation. NOTE: All values in this table are $*10^5$.

arrangements of the symbols *, +, and $) for 4 seconds followed by a blank screen for 4 seconds, followed by a sentence (e.g. "The star is above the plus.") for 4 seconds. Participants could press buttons indicating whether the sentence correctly described the picture at any time during the second stimulus presentation. The participants then rested for 15 seconds before the next trial began. In the other half of the trials the sentence was presented first and the picture second, using the same timing. Figure 1.1 diagrams the temporal structure of the trials.

Imaging was carried out on a 3.0 Tesla G.E. Signa scanner. A T2*-weighted, single-shot spiral pulse sequence was used with TR = 500 ms, TE= 18 ms, 50-degree flip angle. This sequence allowed us to acquire 8 oblique axial slices every 500 ms, with an in-plane resolution of 3.125 millimeters and slice thickness of 3.2 mm, resulting in approximately 5000 voxels per participant. The 8 slices were chosen to cover areas of the brain believed

to be relevant to the task at hand. Each participant was also annotated with anatomical regions of interest (ROIs). The data were preprocessed to remove artifacts due to head motion and signal drift using the FIASCO program Eddy et al. [1999].

5.2.1 Models

Our experiments compare seven different HPMs, and compare them against a standard classifier. In this section, we describe the details of each model.

GNB

The Gaussian Naive Bayes classifier (Mitchell [1997]) used in our experiments operates on the 8 second window following each stimulus presentation. It considers the two windows per trial as two independent examples from one of two classes: either ReadSentence or ViewPicture. For each class, a mean and variance are estimated for each feature independently. In this case, the features are the values of every voxel at every timepoint in the window (16 images, so $16 \times V$ features, where V is the number of voxels being modeled). Once the mean and variance estimates are computed from the training set, they are used to compute the probability of a test example belonging to one class or the other, again under the assumption that all features are independent. The class with the higher probability is predicted for each example, and the classification accuracy for a given fold is the percentage of examples predicted correctly.

There are a few key differences between the GNB approach and HPMs. Firstly, the GNB considers the examples corresponding to the first and second stimuli of each trial in the test set independently (i.e. the N trials become $2N$ separate examples instead of pairs of examples). Through the set of configurations, HPMs are provided with the prior knowledge that in each trial, there must be exactly one instance of ReadSentence and exactly one of ViewPicture. Therefore HPMs cannot make the mistake of predicting two ReadSentence instances for a test trial, whereas the GNB can. Secondly, the GNB has a different noise model from HPMs. The GNB estimates a variance for every voxel at every time point; HPMs estimate a variance for every voxel, independetly of time and independently of process (similar to class).

HPM-GNB

The HPM-GNB model approximates the GNB through the HPM framework. It has two processes: ReadSentence and ViewPicture. The duration of each process is 8 seconds (16 images) so that they may not overlap. HPM-GNB essentially eliminates the differences between HPMs and GNB discussed above. It has a variance for each voxel, and incorporates the knowledge that each trial has exactly one instance of ReadSentence and exactly one of ViewPicture. It is the same as HPM-2-K (presented in the next section) except that the processes are shorter so that they may not overlap.

HPM-2-K and HPM-2-U

HPM-2-K is a 2-process HPM containing just ReadSentence and ViewPicture, each of which has a duration of 12 seconds (a reasonable length for the hemodynamic response function). The 'K' stands for 'known,' meaning that the timing of these two processes is assumed to be known. More specifically, it is assumed that each process starts exactly when its corresponding stimulus is presented. In the notation of Chapter 2, $\Omega = \{0\}$ for both processes. For a given test set trial, the configurations for HPM-2-K are shown in Table 5.2. The configurations for a training set trial are restricted to those containing the correct order of the processes (based on the order of the stimuli).

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$
1	S	1	0	P	17	0
2	P	1	0	S	17	0

Table 5.2: Configurations for a test set trial under HPM-2-K. For supervised training, there is only one configuration for each training trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset.

HPM-2-U is also a 2-process HPM containing just ReadSentence and ViewPicture. In this case though, the 'U' stands for 'unknown,' meaning that we allow a small amount of uncertainty about the start times of the processes. In HPM-2-U, the Ω for each process is $\{0, 1\}$, which translates to delays of 0 or 0.5 seconds (0 or 1 images) following the relevant stimulus presentation. The configurations for the test set in this case are in Table 5.3. Again, during supervised training, the configurations for a given training trial are limited to those that have the correct process ordering. Since the true offset for the processes are unknown even during training, there are still 4 configurations for each training trial.

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$
1	S	1	0	P	17	0
1	S	1	1	P	17	0
1	S	1	0	P	17	1
1	S	1	1	P	17	1
2	P	1	0	S	17	0
2	P	1	1	S	17	0
2	P	1	0	S	17	1
2	P	1	1	S	17	1

Table 5.3: Configurations for a test set trial using HPM-2-U. For supervised training where we assume the order of the stimuli is known, there will be only four configurations for each training trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset.

HPM-3-K and HPM-3-U

HPM-3-K and HPM-3-U add a Decide process to the previous 2 models, also with a duration of 12 seconds. We used a duration of 12 seconds despite reaction times faster than that because we expected that the hemodynamic response would be temporally sluggish compared to the reaction times. In both cases, the Decide process has $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7\}$, allowing the process to start with a delay of 0 to 3.5 seconds from the second stimulus presentation. For these HPMs, we can use the information we have about the participant's reaction time by assuming that the Decide process must begin before the participant pushes the button. This means that for each trial, we can also eliminate any configurations in which the Decide process starts after the button press; that is, any configurations for which $o_3 > reaction\_time$. Note that this implies that different trials can have different sets of configurations since reaction times varied from trial to trial. If the participant did not press the button for some trial, all offsets are considered for the Decide process. The test set configurations for HPM-3-K for a trial with a reaction time of 2.6 seconds are given in Table 5.4. Since the nearest preceding image to the button press corresponds to offset 5, we have removed configurations with $o_3 \in \{6, 7\}$. To do supervised training, we only use the configurations with the correct ordering of ReadSentence and ViewPicture. The configurations for HPM-3-U are the analogous extension to those for HPM-2-U; that is, the configurations in Table 5.4 are extended to allow offsets of 0 or 1 for the first two process instances.

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$	$\pi(i_3)$	$\lambda(i_3)$	$O(i_3)$
1	S	1	0	P	17	0	D	17	0
2	S	1	0	P	17	0	D	17	1
3	S	1	0	P	17	0	D	17	2
4	S	1	0	P	17	0	D	17	3
5	S	1	0	P	17	0	D	17	4
6	S	1	0	P	17	0	D	17	5
7	P	1	0	S	17	0	D	17	0
8	P	1	0	S	17	0	D	17	1
9	P	1	0	S	17	0	D	17	2
10	P	1	0	S	17	0	D	17	3
11	P	1	0	S	17	0	D	17	4
12	P	1	0	S	17	0	D	17	5

Table 5.4: Configurations for a test set trial under HPM-3-K. The reaction time for this trial is 2.6 seconds, which corresponds to offset 5 for the Decide process, so all configurations with offsets greater than 5 for this process have been eliminated for this trial. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset.

HPM-4-K and HPM-4-U

The final two rows of Table 5.10 show results for HPMs similar to HPM-3-K and HPM-3-U called HPM-4-K and HPM-4-U. These HPMs include another process called PressButton, with a duration of 12 seconds. The landmark for instances of this process is the time at which the participant pressed the button rounded to the nearest preceding image. The offsets are $\Omega = \{-1, 0\}$, indicating that the process may begin 0 or 0.5 seconds before the landmark. If the participant did not press the button during a particular trial, this process is not included in the configurations for that trial. That is, we are treating this process as observable in terms of its existence or non-existence, with a small amount of uncertainty in its timing when it does exist. Furthermore, we can assume that Decide must begin no later than PressButton, and remove configurations in which this assumption does not hold. The configurations for HPM-4-K for a test trial in which the participant pressed the button 1.75 seconds (between offsets 3 and 4) after the second stimulus presentation are shown in Table 5.5. If there was no button press, the configurations for HPM-4-K are identical to those for HPM-3-K (and those for HPM-4-U match the set for HPM-3-U). For supervised training, we simply limit the configurations to those with ReadSentence and ViewPicture in the correct order.

c	$\pi(i_1)$	$\lambda(i_1)$	$O(i_1)$	$\pi(i_2)$	$\lambda(i_2)$	$O(i_2)$	$\pi(i_3)$	$\lambda(i_3)$	$O(i_3)$	$\pi(i_4)$	$\lambda(i_4)$	$O(i_4)$
1	S	1	0	P	17	0	D	17	0	B	3	-1
2	S	1	0	P	17	0	D	17	0	B	3	0
3	S	1	0	P	17	0	D	17	1	B	3	-1
4	S	1	0	P	17	0	D	17	1	B	3	0
5	S	1	0	P	17	0	D	17	2	B	3	-1
6	S	1	0	P	17	0	D	17	2	B	3	0
7	S	1	0	P	17	0	D	17	3	B	3	0
8	P	1	0	S	17	0	D	17	0	B	3	-1
9	P	1	0	S	17	0	D	17	0	B	3	0
10	P	1	0	S	17	0	D	17	1	B	3	-1
11	P	1	0	S	17	0	D	17	1	B	3	0
12	P	1	0	S	17	0	D	17	2	B	3	-1
13	P	1	0	S	17	0	D	17	2	B	3	0
14	P	1	0	S	17	0	D	17	3	B	3	0

Table 5.5: Configurations for a test set trial using HPM-4-K. For this trial, the participant's reaction time was 1.75 seconds, so the landmark for PressButton (B) is 3 (the nearest preceding image). We have eliminated configurations in which Decide (D) begins after PressButton. For each configuration c, i_k is the kth process instance, $\pi(i_k)$ is its process ID, $\lambda(i_k)$ is its landmark, and $O(i_k)$ is its offset.

5.2.2 Comparing HPMs

Given that we are studying datasets with an unknown number of processes, each of which may have unknown properties, how can we compare a set of HPMs, each describing a different model of the data, against each other? Below we explore two metrics: accuracy in predicting aspects of the dataset that we do know (like the order of the stimuli), and the log-likelihood of the data under the model. Both metrics are computed within the framework of cross-validation.

For each metric, we first report an experiment using all the HPMs described above trained using the standard parameterization on all available voxels for each participant. Since this is computationally expensive, we then restricted our experiments to the top 1000 most active voxels for each participant and the subset of HPMs with known timing for the first two processes. (It does seem reasonable to assume that the start times of these two processes coincide with their corresponding stimulus presentations.) These experiments compare standard HPMs, regularized HPMs, and HPMs with basis functions.

In the regularization experiments, the temporal smoothness and spatial smoothness penalties described in Chapter 3 were weighted by a regularization parameter chosen using HPM-3-K on Participant A. From the set of powers of two (with exponents from 0 to 10), the value with the highest average test set log-likelihood over baseline was at 512. We used the temporal smoothness and spatial smoothness penalties to reflect our prior knowledge that the hemodynamic response is based on blood flow, which should be smooth over space and time. We chose not to use the spatial sparsity penalty because the experiments were already limited to 1000 voxels, and we did not have sufficient prior knowledge to use the spatial prior penalty.

The HPMs parameterized with basis functions used the basis set shown in Figure 5.8. These basis functions were generated using the method described in Section 3.2, based on the work in Hossein-Zadeh et al. [2003]. To generate this basis, we first created a matrix $\mathbf{Q}$ containing 10,000 realizations over 24 time points of the gamma function:

$$h(t) = \exp(-\frac{t}{\sqrt{ab}})\frac{et\sqrt{b/a}}{b}$$

where the parameter a ranged from 0.05 to 0.21 and the parameter b ranged from 3 to 7. The basis set is the first three principal components of $\mathbf{Q}'\mathbf{Q}$.

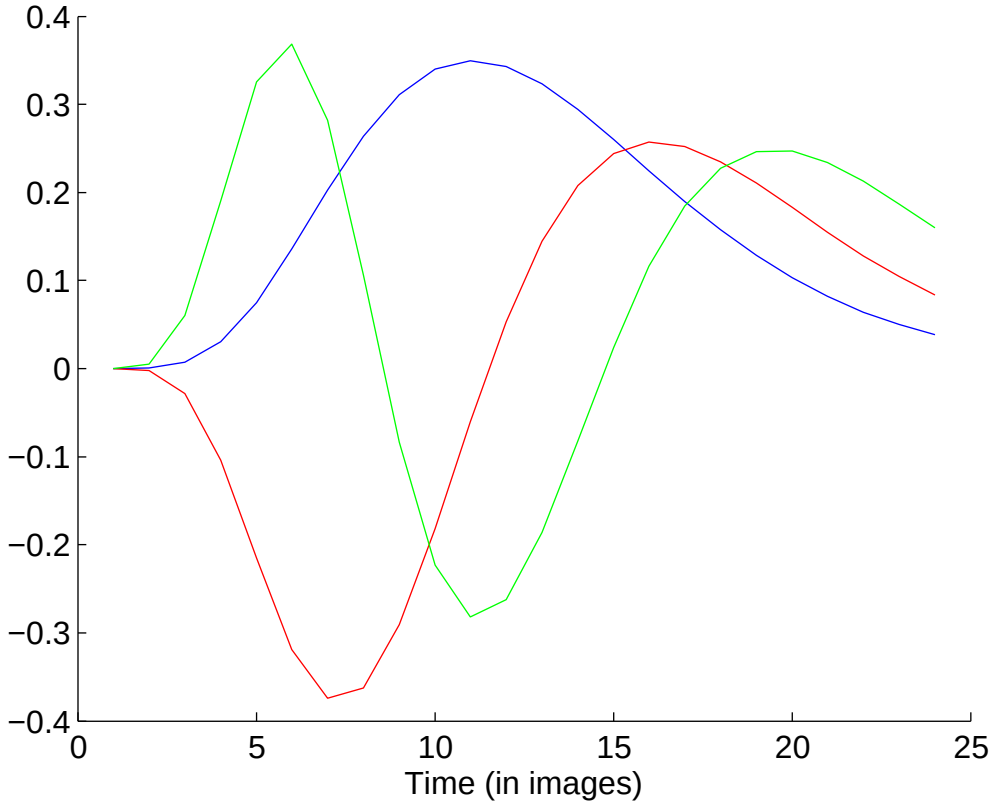

Figure 5.8: Basis functions used to parameterize process response signatures in sentence-picture experiments.

Data Log-Likelihood

To compute the log-likelihood of a single trial of data with respect to a given HPM, we compute the log-likelihood of the trial under each test set configuration and weight that log-likelihood by the posterior probability of the configuration. Since in practice, the posterior probability distributions tend to be sharply peaked, almost always assigning probability 1 to a single configuration, this can be thought of as the log-likelihood of the maximum a posteriori (MAP) configuration for each trial. For each configuration, we compute a predicted trial by adding the response signature of each process instance in the configuration to the window of the trial in which it is active. The average trial (from the training set) is used to fill in any time points in the predicted trial for which no process instance is active. This predicted trial is the mean used for the log-likelihood computation.

Since log-likelihood has no concrete units, we also computed a baseline log-likelihood from a naive method, and we compare performance across models and participants using improvement over this baseline. The naive method used to create the baseline is to simply average the time courses of the training trials without regard to processes, and use that time course as the mean for the test trials. Note that this baseline averages together trials with all possible orders of the stimuli (picture followed by sentence and sentence followed by picture).

Table 5.6 reports the average improvement over baseline test set log-likelihood over 5 folds of cross-validation for each of 13 participants under seven different HPMs. Table 5.7 reports the same metric for the top 1000 most active voxels for each participant under the restricted set of four HPMs. Tables 5.8 and 5.9 give the corresponding results under the same experimental conditions as Table 5.7, but for regularized and basis function HPMs, respectively.

The first result of note is that for the standard HPMs (Tables 5.6 and 5.7) and for the vast majority of participant-HPM combinations, the improvement in log-likelihood over the baseline is a negative number. That is, the data log-likelihood is higher using the baseline method of predicting the mean training trial than when using HPMs. These results indicate that standard HPMs are overfitting the training data. The presence of overfitting is unsurprising given the ratio of parameters to training data. (Recall that each process in an HPM has DV parameters to characterize its response signature (e.g. $DV = 24 \times 1000$), and only N trials to use as examples (in 5-fold cross-validation over 40 trials, $N = 32$). This is in addition to the parameters of the processes' timing distributions and the V process-independent noise parameters.) The scores get worse as the HPMs become more complex, providing more evidence for overfitting.

Table 5.8 suggest that regularization is able to mitigate overfitting. All of the improve-

ments in data log-likelihood over the baseline are positive, and the improvements now increase with the complexity of the HPM. As a general trend, it appears that HPM-3-K and HPM-4-K outperform HPM-GNB and HPM-2-K, but the differences between HPM-3-K and HPM-4-K are small. The trends for HPMs with basis functions (Table 5.9) are very similar to those for regularized HPMs. On average, the regularized HPMs just barely outperform HPMs with basis functions. Both regularization and basis functions provide HPMs with a temporal smoothness constraint, which appears to help them both outperform standard HPMs. Regularization also provides a spatial smoothness constraint, which is not present in the basis function version of HPMs. In contrast, basis functions provide a significant decrease in the number of parameters in the model. The fact that regularized HPMs have slightly higher performance than basis function HPMs may suggest that the spatial smoothness constraint is more important than the parameter reduction. Another possible explanation for the lower performance of HPMs with basis functions is that the basis set used here might not be expressive enough to capture the correct process response signatures.

Participant	GNB	HPM- 2-K	2-U	3-K	3-U	4-K	4-U
A	0.162±0.29	-0.487±0.059	-0.178±0.13	-1.24±0.21	-1.03±0.26	-2.01±0.16	-1.75±0.20
B	0.217±0.22	-0.196±0.12	-0.102±0.066	-1.12±0.23	-1.07±0.21	-2.17±0.15	-2.03±0.18
C	-0.0940±0.33	-0.475±0.17	-0.360±0.14	-1.32±0.27	-1.18±0.26	-2.74±0.24	-2.47±0.35
D	-0.0677±0.13	-0.326±0.096	-0.264±0.16	-1.23±0.13	-1.11±0.12	-2.60±0.37	-2.47±0.34
E	0.0326±0.16	-0.443±0.13	-0.396±0.18	-1.62±0.11	-1.50±0.14	-2.70±0.37	-2.48±0.26
F	0.325±0.35	-0.183±0.30	0.0186±0.38	-1.34±0.17	-1.02±0.13	-2.66±0.84	-2.44±0.86
G	0.0652±0.055	-0.311±0.071	-0.244±0.081	-1.31±0.089	-1.14±0.15	-2.00±0.27	-1.89±0.20
H	0.117±0.096	-0.328±0.093	-0.264±0.12	-1.28±0.13	-1.27±0.063	-2.24±0.32	-2.01±0.32
I	0.00216±0.21	-0.322±0.23	-0.261±0.22	-1.14±0.19	-1.03±0.15	-2.14±0.30	-2.02±0.30
J	-0.171±0.16	-6.78±0.16	-4.26±0.099	-1.29±0.19	-1.06±0.18	-2.50±0.31	-2.28±0.37
K	0.0357±0.24	-0.273±0.16	-0.169±0.23	-1.15±0.24	-1.01±0.20	-2.55±0.43	-2.36±0.56
L	0.228±0.20	-3.95±0.058	-3.20±0.090	-1.73±0.064	-1.64±0.049	-3.41±0.68	-3.26±0.60
M	0.0445±0.11	-0.375±0.19	-0.330±0.22	-1.20±0.12	-1.15±0.13	-2.42±0.11	-2.22±0.11
Mean	0.0690	-1.11	-0.770	-1.31	-1.17	-2.47	-2.28

Table 5.6: Improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). HPM-N-U is an HPM with N processes with unknown timing for ReadSentence and ViewPicture (the processes may begin 0 or 1 images after the corresponding stimulus is presented).

Participant	HPM- GNB	2-K	3-K	4-K
A	-0.0533±0.11	-0.224±0.038	-0.163±0.077	-0.327±0.063
B	0.0536±0.12	0.0251±0.054	-0.144±0.077	-0.392±0.058
C	-0.116±0.15	-0.802±0.067	-0.236±0.065	-0.488±0.097
D	-0.0924±0.059	0.0192±0.069	-0.0952±0.061	-0.375±0.14
E	-0.0710±0.053	-0.117±0.045	-0.275±0.021	-0.492±0.062
F	0.0924±0.15	0.0155±0.15	-0.196±0.068	-0.445±0.18
G	-0.00969±0.025	-0.0580±0.036	-0.260±0.032	-0.404±0.099
H	0.00556±0.043	-0.0358±0.045	-0.201±0.051	-0.466±0.079
I	-0.0133±0.071	-0.0531±0.069	-0.229±0.048	-0.511±0.049
J	-0.126±0.095	-0.165±0.091	-0.207±0.019	-0.477±0.071
K	-0.0654±0.12	0.0304±0.11	-0.106±0.13	-0.402±0.13
L	0.0267±0.056	-0.0817±0.023	-0.279±0.031	-0.574±0.098
M	-0.0124±0.074	-0.0506±0.099	-0.210±0.089	-0.482±0.071
Mean	-0.0293	-0.115	-0.200	-0.449

Table 5.7: For the baseline HPM with the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

Participant	HPM-			
	GNB	2-K	3-K	4-K
A	0.194±0.15	0.170±0.078	0.525±0.12	0.525±0.16
B	0.357±0.12	0.492±0.073	0.564±0.092	0.558±0.096
C	0.181±0.17	0.388±0.096	0.478±0.086	0.483±0.10
D	0.193±0.070	0.459±0.082	0.600±0.041	0.610±0.065
E	0.211±0.080	0.327±0.079	0.413±0.064	0.371±0.049
F	0.344±0.16	0.417±0.18	0.442±0.095	0.369±0.11
G	0.265±0.035	0.376±0.049	0.408±0.55	0.397±0.070
H	0.309±0.042	0.441±0.055	0.516±0.067	0.512±0.067
I	0.289±0.086	0.413±0.095	0.473±0.090	0.462±0.082
J	0.185±0.082	0.314±0.087	0.506±0.071	0.520±0.084
K	0.231±0.13	0.489±0.14	0.587±0.17	0.579±0.17
L	0.315±0.078	0.373±0.038	0.427±0.051	0.362±0.052
M	0.289±0.067	0.424±0.092	0.503±0.089	0.507±0.097
Mean	0.259	0.391	0.496	0.481

Table 5.8: For HPMs with spatial smoothness and temporal smoothness regularization penalties on the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

Participant	HPM-			
	GNB	2-K	3-K	4-K
A	0.0465±0.091	0.0371±0.093	0.343±0.29	0.383±0.22
B	0.318±0.075	0.486±0.067	0.531±0.090	0.543±0.062
C	0.0653±0.083	0.352±0.066	0.416±0.051	0.472±0.014
D	0.0693±0.15	0.371±0.13	0.477±0.12	0.514±0.10
E	0.0535±0.076	0.254±0.082	0.325±0.073	0.315±0.047
F	0.395±0.31	0.476±0.33	0.560±0.31	0.479±0.23
G	0.224±0.089	0.384±0.092	0.405±0.097	0.415±0.093
H	0.328±0.056	0.471±0.078	0.596±0.33	0.608±0.060
I	0.275±0.10	0.413±0.10	0.486±0.10	0.471±0.094
J	0.148±0.085	0.327±0.088	0.482±0.088	0.511±0.063
K	0.184±0.19	0.469±0.15	0.530±0.14	0.533±0.20
L	0.279±0.069	0.401±0.072	0.491±0.090	0.452±0.064
M	0.223±0.10	0.417±0.12	0.483±0.11	0.504±0.11
Mean	0.201	0.374	0.471	0.477

Table 5.9: For HPMs parameterized with basis functions using the top 1000 most active voxels, improvement over baseline of log-likelihood, $\pm$ one standard deviation, of the 8-trial test set under the HPM trained on the remaining 32 trials averaged over 5-fold cross-validation for 13 participants. All values are $\times 10^4$. The baseline is the log-likelihood of the data obtained using the mean of all the training trials to predict each test trial, ignoring stimulus order and participant response. Note that negative values mean that the baseline outperformed the model. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

Classification Accuracy

As an alternative to cross-validated data log-likelihood, we consider here using classification accuracy to compare HPMs. First, we briefly describe how the classification accuracy for a single fold of the cross-validation is computed for the HPM models. On a given fold, each HPM contains a set of configurations for each of the 32 training trials and a set of configurations for each of the 8 test trials. We estimate the parameters of the HPM using the training set (using the learning algorithms in Chapter 2) and use them to compute the posterior probability of each test set configuration for each test trial (using the inference algorithm in Chapter 2). Given the posterior probabilities of the configurations for a particular test trial, we can compute the marginal probability of any property of the configurations. For example, we can sum the posterior probabilites of all the configurations whose first process instance is ViewPicture to get the marginal probability that the first process instance of this test trial is ViewPicture. In fact, we compute these marginal probabilities for all combinations of process instance and process ID independently. We can then make a prediction by choosing the process ID with the highest marginal probability for each process ID. For each trial, this prediction is compared with the true process IDs. The fraction correct on each trial is averaged over trials to obtain the classification accuracy over the whole test set. This is repeated for each fold of the cross-validation, and those results are once again averaged to produce a single classification accuracy. The standard deviations reported in Table 5.10 are computed across folds of the cross-validation using this mean.

Table 5.10 presents 5-fold cross-validated accuracies for the task of classifying ReadSentence and ViewPicture processes for eight different models in all 13 participants, using standard HPMs. Table 5.11 reports the same accuracies, but restricted to the top 1000 most active voxels for each participant, and the smaller set of models. Tables 5.12 and 5.13 give the corresponding results for HPMs with regularization and basis functions, respectively.

The first general result of note involving classification accuracies is that on average, HPM-2-K performs slightly better than HPM-GNB, but adding the third and fourth processes to the model tends to degrade performance. That HPM-2-K improves upon HPM-GNB suggests that modeling the temporal overlap between the first two processes' response signatures is helpful. The decreased performance for the more complex HPMs may or may not be surprising. On the one hand, we might expect that adding the Decide process would improve classification accuracies among the first two processes by deconvolving the beginning of the third processes from the end of the second process. However, this assumes that the Decide processes begin soon enough after the second stimuli to have an impact on the estimation of ViewPicture and ReadSentence, and that the response

signature of the third process has significant spatial overlap with the first two. If these assumptions are not met, the addition of a third process could be detrimental because of the additional parameters that the learning algorithm to estimate, taking focus away from the two processes involved in the classification task. In our experiments, it appears that the potential benefits of deconvolving contributions of the third process from the first two processes is outweighed by the additional parameters required to add the Decide process.

Another general trend in the results reporting classification accuracies is that none of the HPMs beat the classification scores of a Gaussian Naive Bayes classifier. In considering this result, it is important to recall that GNB has a different noise model than HPMs. GNB has a feature for each voxel at each image (time point) of the 8 seconds following the stimulus. For each feature, GNB estimates a mean and variance parameter for each class. In contrast, HPMs have process-independent (analagous to class-independent) noise parameters for each voxel. GNB and HPM-GNB provide a direct comparison of the two noise models, showing that the class-specific noise model in GNB is advantageous. While HPMs do not classify the stimuli as well as GNB, they do tend to perform better than random in most cases. (Since this is a binary classification task, the null hypothesis is that for each fold, the number of correct classifications achieved by a classifier with no discriminative power will be Binomial with parameters $p = 0.5$ and $N = 8$. At significance level $\alpha = 0.05$, we would reject this null hypothesis for $X \geq 6$ correct classifications, or accuracy of 0.75.) This suggests that even though HPMs are not optimized to find the decision boundary between ViewPicture and ReadSentence, they do learn information that discriminates between these two processes to some extent.

Finally, we note that on average, HPMs parameterized with basis functions slightly outperform regularized HPMs, which outperform the standard HPMs in terms of classification accuracy on this task. This is in contrast to the comparison in terms of held-out data log-likelihood, where regularized HPMs outperformed HPMs with basis functions on average. The different metrics lend themselves to slightly different interpretations. The log-likelihood metric speaks to the ability of the HPM in question to track the time series, whereas the classification metric addresses the ability of the HPM to model the decision boundary of the classification task. The metrics are related through the posterior distribution on configurations. The log-likelihood metric is a weighted average of the log-likelihood under each configuration weighted by its posterior probability, which in practice generally corresponded to the log-likelihood under the maximum a posteriori configuration. The classification metric computes the marginal probabilities for the classification task from the posterior distribution over configurations, which also generally corresponded to the classification from the MAP configuration. HPMs are optimized in training for log-likelihood, so we suggest using the log-likelihood metric as the primary

evaluation criterion, and using the classification accuracy secondarily.

Participant	GNB	HPM-GNB	HPM-2-K	HPM-2-U	HPM-3-K	HPM-3-U	HPM-4-K	HPM-4-U
A	80.0±16.2	62.5±17.7	67.5±11.2	62.5±17.7	65.0±20.5	72.5±20.5	75.0±14.4	64.2±15.8
B	92.5±8.2	92.5±11.2	97.5±5.6	97.5±5.6	95.0±6.9	95.0±6.9	93.3±5.6	90.8±11.2
C	95.0±8.1	82.5±14.3	82.5±14.2	75.0±17.7	72.5±16.3	77.5±10.5	74.2±9.0	69.2±14.0
D	100.0±0.0	92.5±6.8	97.5±5.6	100.0±0.0	87.5±8.8	87.5±8.8	74.2±12.3	80.0±7.5
E	95.0±5.2	87.5±8.8	77.5±10.5	70.0±20.9	80.0±16.8	75.0±12.5	79.2±19.3	81.7±21.4
F	73.8±19.5	80.0±11.2	75.0±8.8	80.0±11.2	82.5±14.3	72.5±10.5	80.0±8.5	72.5±10.5
G	93.8±4.4	70.0±19.0	62.5±17.7	70.0±6.8	67.5±16.8	60.0±10.5	84.2±10.4	80.8±7.6
H	87.5±9.9	90.0±10.5	82.5±6.8	90.0±10.5	82.5±14.3	85.0±10.5	82.5±17.0	79.2±6.6
I	88.8±12.0	82.5±6.8	82.5±11.2	80.0±16.8	75.0±15.3	75.0±17.7	80.8±12.7	76.7±12.0
J	93.8±4.4	80.0±11.2	70.0±16.8	75.0±19.8	82.5±19.0	85.0±16.3	78.3±16.0	73.3±16.3
K	96.3±8.4	75.0±19.8	90.0±16.3	87.5±21.7	87.5±15.3	85.0±16.3	81.7±14.0	81.7±14.0
L	71.3±23.6	57.5±16.8	62.5±15.3	65.0±20.5	67.5±20.9	52.5±18.5	52.5±19.5	57.5±14.8
M	97.5±3.4	80.0±14.3	82.5±14.3	82.5±14.3	67.5±16.8	70.0±19.0	83.3±13.5	80.0±14.3
Mean	89.6	79.4	72.9	79.6	77.9	76.3	78.4	76.0

Table 5.10: Percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented). HPM-N-U is an HPM with N processes with unknown timing for ReadSentence and ViewPicture (the processes may begin 0 or 1 images after the corresponding stimulus is presented).

Participant	GNB	HPM- GNB	HPM- 2-K	HPM- 3-K	HPM- 4-K
A	86.3±14.3	67.5±6.8	67.5±16.8	67.5±16.8	67.5±16.8
B	92.5±10.3	97.5±5.6	97.5±5.6	97.5±5.6	90.8±11.2
C	97.5±5.6	85.0±16.3	90.0±16.3	90.0±16.3	85.0±16.3
D	97.5±3.4	95.0±11.2	97.5±5.6	97.5±5.6	85.8±7.0
E	100.0±0.0	97.5±5.6	92.5±11.2	85.0±16.3	89.2±12.0
F	81.3±21.7	80.0±6.8	85.0±10.5	85.0±5.6	77.5±12.7
G	97.5±5.6	77.5±10.5	72.5±13.7	80.0±14.3	81.7±8.1
H	96.3±5.6	92.5±11.2	95.0±6.8	90.0±6.8	88.3±4.6
I	92.5±6.8	87.5±8.8	85.0±16.3	85.0±13.7	88.3±12.3
J	96.3±3.4	92.5±11.2	95.0±11.2	95.0±6.8	88.3±11.6
K	93.8±8.8	82.5±16.8	92.5±16.8	92.5±16.8	86.7±16.2
L	78.8±14.4	67.5±14.3	70.0±14.3	77.5±16.3	69.2±18.5
M	100.0±0.0	92.5±6.8	92.5±11.2	85.0±16.3	91.7±5.9
Mean	93.1	85.8	87.1	86.7	83.8

Table 5.11: For the baseline HPM with the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

Participant	GNB	HPM- GNB	2-K	3-K	4-K
A	86.3±14.3	75.0 ±8.8	70.0±19.0	75.0±17.7	70.0±18.3
B	92.5±10.3	97.5±5.6	97.5±5.6	95.0±11.2	90.8±11.2
C	97.5±5.6	85.0±16.3	95.0±11.2	92.5±16.8	90.0±13.7
D	97.5±3.4	95.0±11.2	100.0±0.0	97.5±5.6	85.8±7.0
E	100.0±0.0	100.0±0.0	92.5±11.2	95.0±6.8	89.2±8.1
F	81.3±21.7	80.0±6.8	85.0±10.5	80.0±14.3	76.7±15.2
G	97.5±5.6	85.0±10.5	77.5±18.5	77.5±16.3	78.3±6.8
H	96.3±5.6	95.0±6.8	97.5±5.6	90.0±5.6	88.3±4.6
I	92.5±6.8	77.5±10.5	90.0±10.5	85.0±10.5	85.8±7.6
J	96.3±3.4	92.5±6.8	97.5±5.6	95.0±6.8	93.3±6.3
K	93.8±8.8	80.0±19.0	92.5±16.8	92.5±16.8	85.0±19.9
L	78.8±14.4	67.5±14.3	77.5±16.3	80.0±14.3	70.8±16.4
M	100.0±0.0	95.0±6.8	97.5±5.6	97.5±5.6	93.3±5.6
Mean	93.1	86.5	90.0	88.7	84.4

Table 5.12: For HPMs with spatial and temporal smoothness regularization penalties on the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

Participant	GNB	HPM- GNB	HPM- 2-K	HPM- 3-K	HPM- 4-K
A	86.3±14.3	72.5±13.7	72.5±20.5	72.5±13.7	82.5±9.9
B	92.5±10.3	97.5±5.6	97.5±5.6	92.5±11.2	88.3±11.2
C	97.5±5.6	90.0±13.7	87.5±12.5	92.5±11.2	92.5±7.5
D	97.5±3.4	100.0±0.0	100.0±0.0	97.5±5.6	85.8±7.0
E	100.0±0.0	100.0±0.0	95.0±11.2	100.0±0.0	94.2±6.3
F	81.3±21.7	80.0±11.2	90.0±5.6	85.0±5.6	79.2±8.3
G	97.5±5.6	82.5±14.3	80.0±11.2	82.5±14.3	79.2±7.8
H	96.3±5.6	97.5±5.6	97.5±5.6	92.5±6.8	93.3±3.7
I	92.5±6.8	85.0±13.7	87.5±8.8	85.0±10.5	80.8±10.5
J	96.3±3.4	95.0±6.8	97.5±5.6	97.5±5.6	90.8±5.4
K	93.8±8.8	92.5±11.2	90.0±16.3	90.0±16.3	86.7±11.9
L	78.8±14.4	82.5±19.0	87.5±8.8	87.5±8.8	69.2±23.9
M	100.0±0.0	100.0±0.0	95.0±11.2	100.0±0.0	97.5±2.3
Mean	93.1	90.4	90.6	90.4	86.2

Table 5.13: For HPMs parameterized with basis functions using the top 1000 most active voxels, percent accuracy ± one standard deviation at classifying the first two process instances per trial as either ReadSentence or ViewPicture averaged over 5-fold cross-validation for 13 participants. GNB is the traditional Gaussian Naive Bayes classifier. HPM-GNB is a Gaussian Naive Bayes with the same noise model and assumptions as HPMs. HPM-N-K is an HPM with N processes with known timing for ReadSentence and ViewPicture (both processes are forced to begin when the corresponding stimulus is presented).

5.2.3 Visualizing Learned HPMs

The above section used 5-fold cross-validation to estimate the performance of a variety of models in terms of classification accuracy and data log-likelihood. Of course, the best model we can learn uses all available data. For this section, we trained a variety of models using all available trials and voxels.

We can visualize the resulting HPMs in a number of ways. To see the spatial response for a process, we can average the response for each voxel over time and plot the result over the whole brain. Examples of this type of visualization are given in Figures 5.9 and 5.10. These figures were extracted from HPM 'browsers' written in Matlab. Appendix F contains images of the processes of HPM-3-K learned using standard, regularized, and basis function versions of HPMs on Participant D (selected for generally high performance across metrics and algorithms). At this level of aggregation, the differences between the three versions of HPMs are almost impossible to see.

Another way to visualize the learned HPMs is to plot the temporal response for a process in any given voxel. Figures 5.11, 5.12, and 5.13 show the time course of a single voxel in the right dorso-lateral prefrontal cortex for standard, regularized, and basis function versions of HPMs respectively, again for Participant D. Unlike the images that are averaged over time, these plots plainly show differences among the methods, with the temporal smoothness of the time course increasing from standard to regularized to basis function HPMs.

We can also compare the learned timing distributions for the Decide process across methods. These results (again for Participant D) are shown in Tables 5.14, 5.15, and 5.16, for standard, regularized, and basis function versions of HPMs respectively. The standard and regularized HPMs have very similar timing distributions for Decide, whereas the distribution for HPMs with basis functions is skewed more toward the beginning of the interval.

These visualizations are important for interpretating the learned HPMs, and being able to visualize them at all for a process with an onset that is both unknown and variable from one trial to the next is a major contribution of HPMs. Recall that the true meaning of this process is unsupervised; it is the process learned from the training set that begins between the second stimulus and the button press. To fully understand the process, we must examine the process response signature and timing distribution in the context of the study.

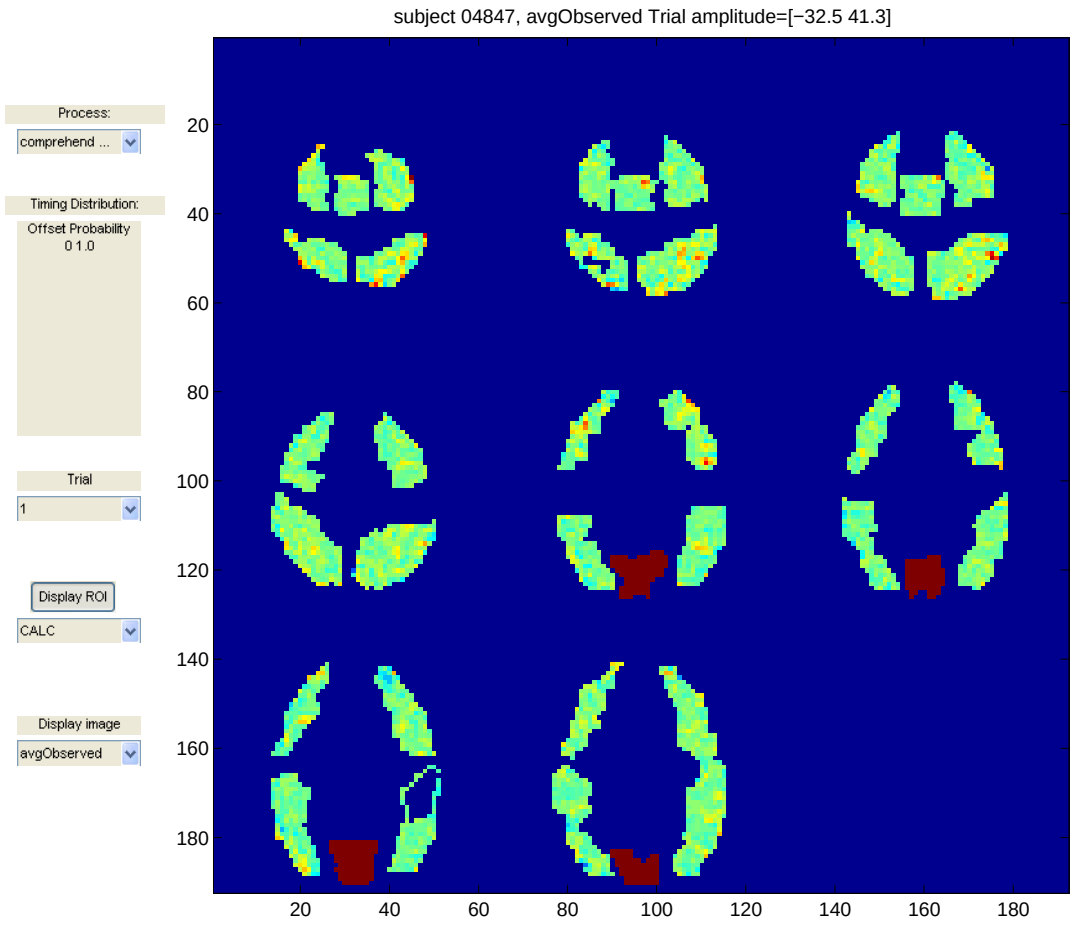

Figure 5.9: Screen shot of an HPM viewer written in Matlab. Drop-down menus select from the following views: observed data (for any particular trial), observed data averaged over all trials, predicted data (for any particular trial), and process response signatures (for any particular process). Anatomical regions of interest (ROIs) can also be highlighted, as demonstrated here for the visual cortex.

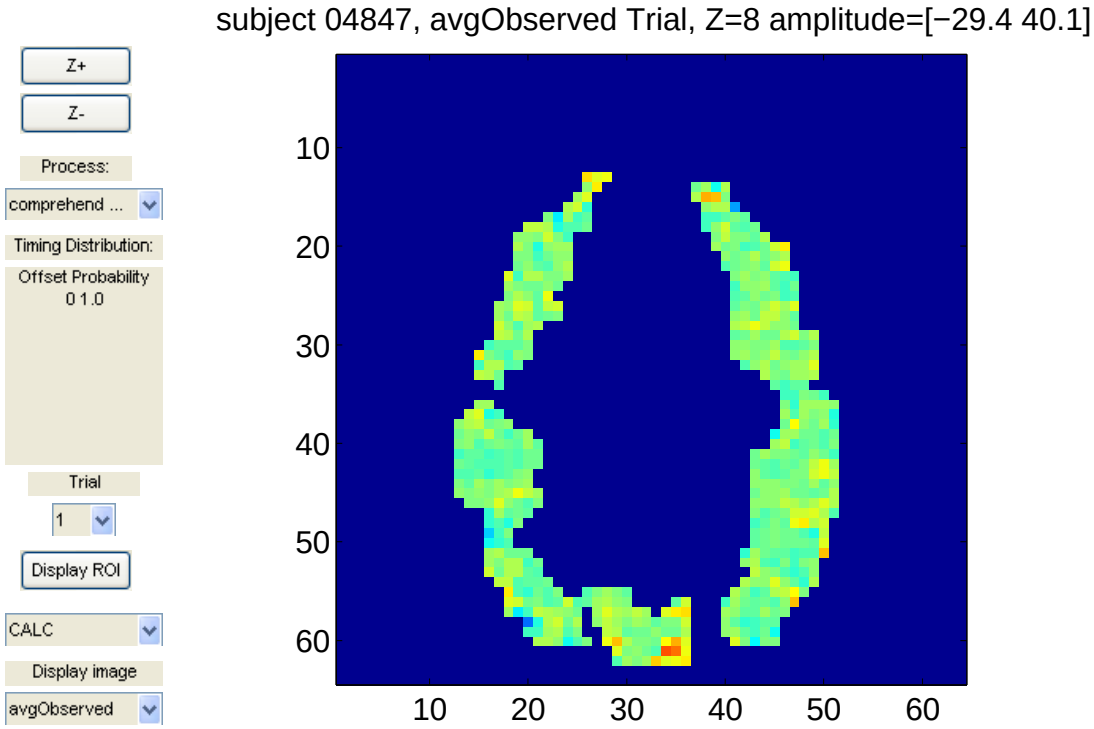

Figure 5.10: Screen shot of an HPM viewer written in Matlab. The viewer displays a single z-slice at a time, and clicking on a voxel brings up a new plot of the temporal response for that voxel (which was averaged to produce the original plot). Drop-down menus select from the following views: observed data (for any particular trial), observed data averaged over all trials, predicted data (for any particular trial), and process response signatures (for any particular process). Anatomical regions of interest (ROIs) can also be highlighted.

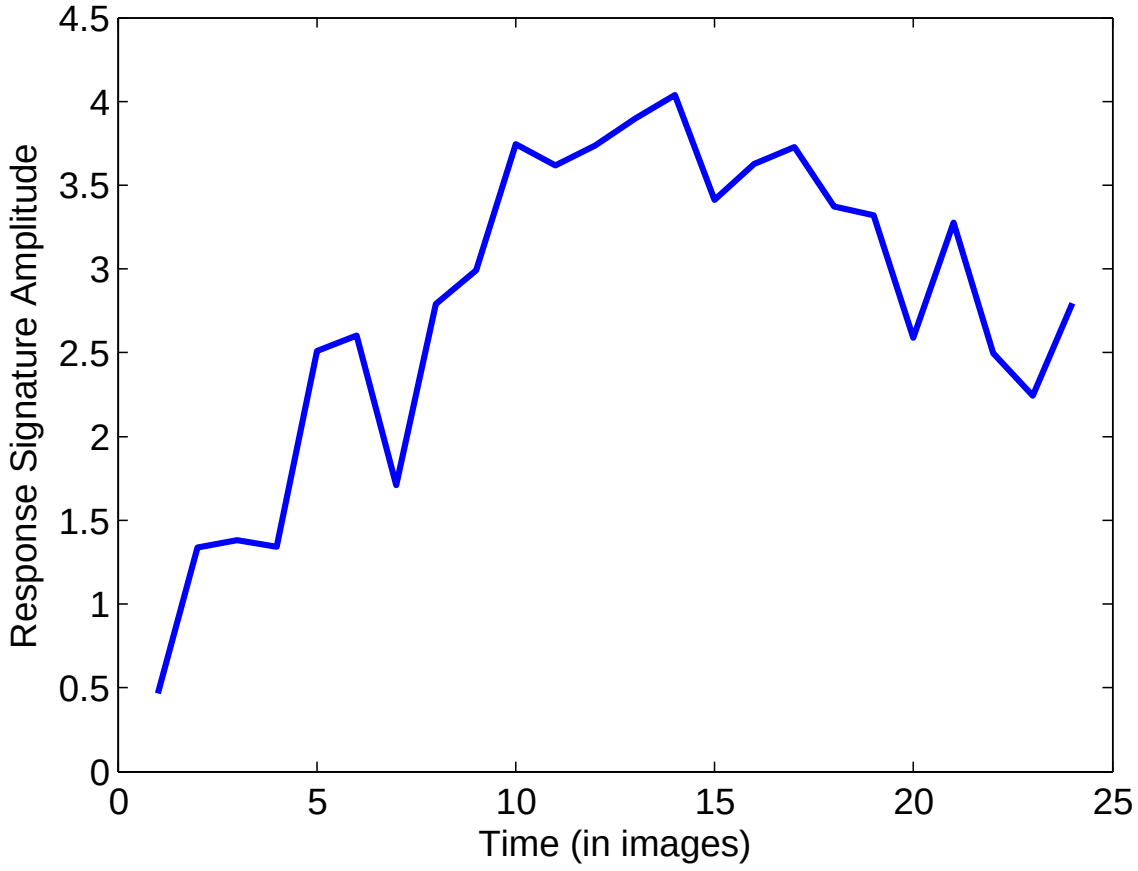

Figure 5.11: The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5).

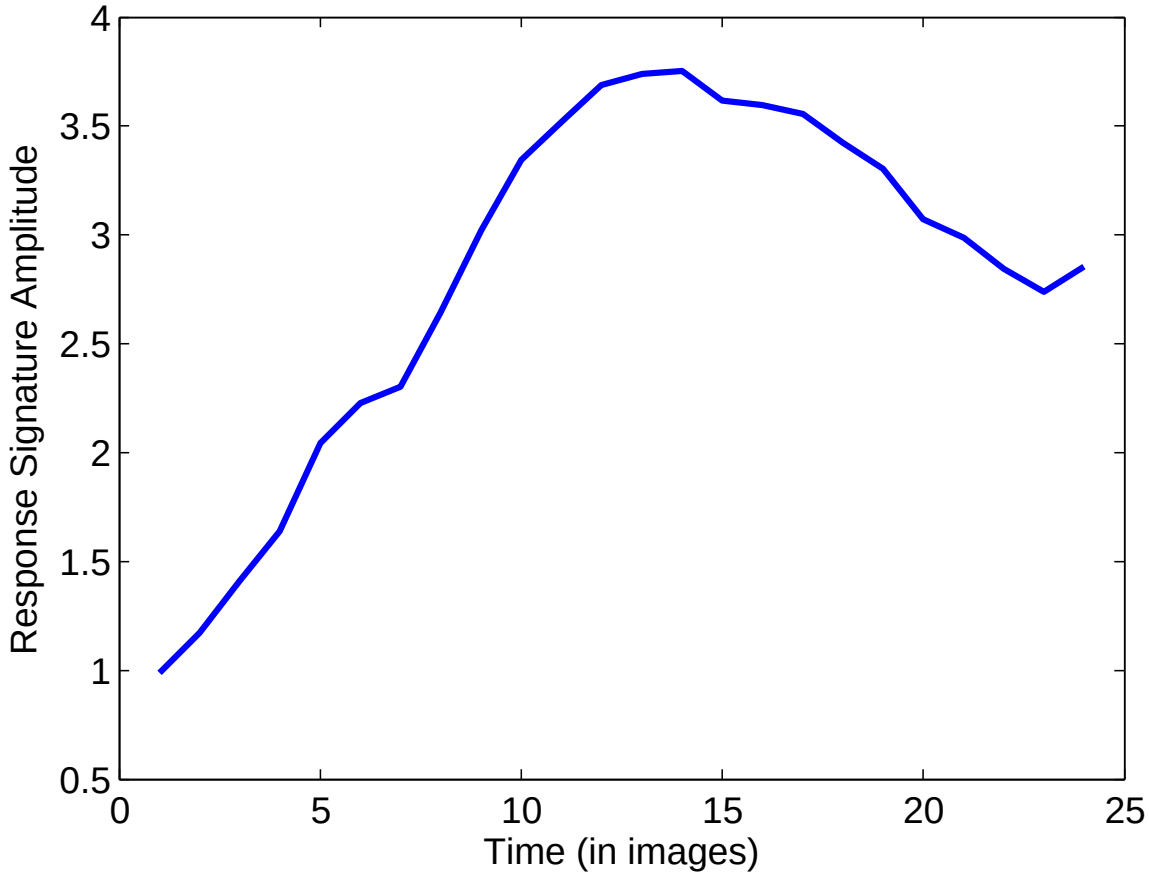

Figure 5.12: The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using regularized HPMs for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5).

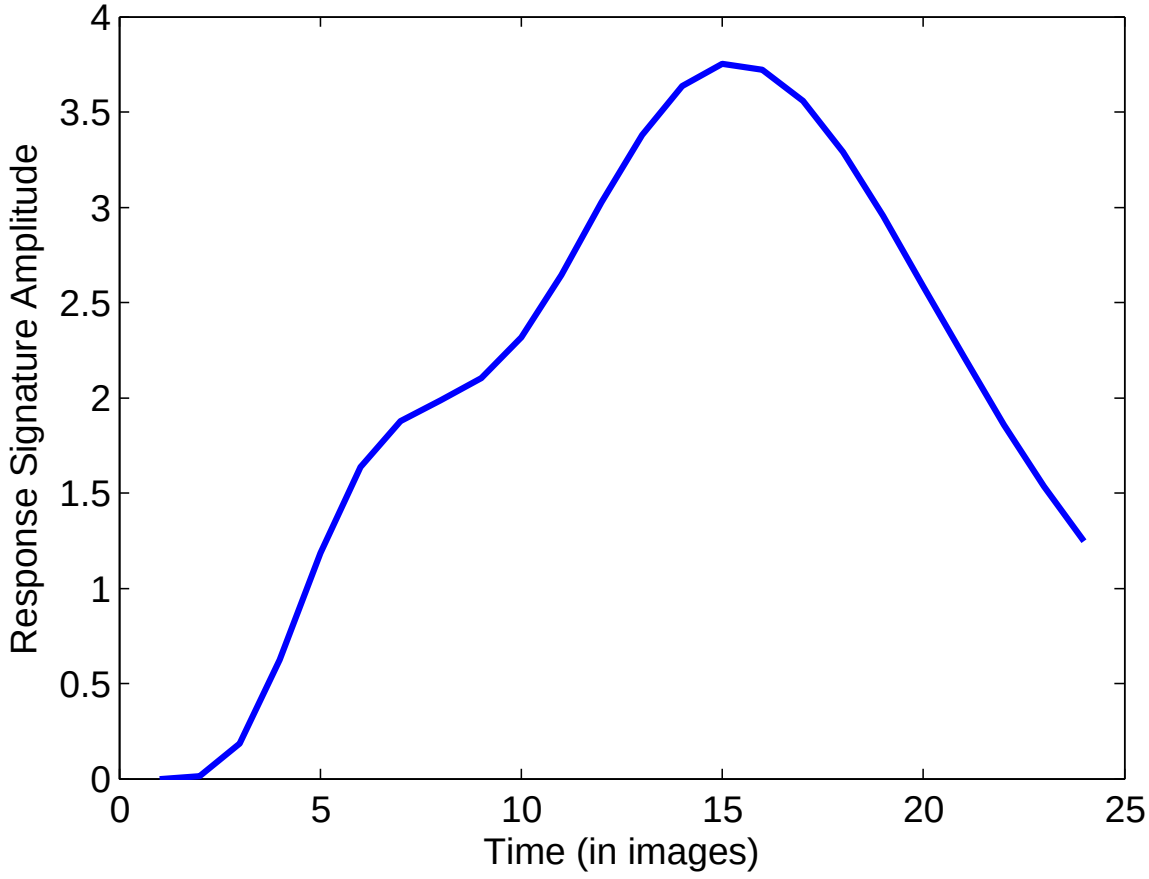

Figure 5.13: The timecourse of the response signature for the Decide process in HPM-3-K learned from all trials and all voxels using HPMs with basis functions for one voxel in the right dorsolateral prefrontal cortex (in z-slice 5).

Offset:	0	1	2	3	4	5	6	7
Theta:	0.2957	0.0762	0.1006	0.0518	0.0518	0.1982	0.0762	0.1494

Table 5.14: Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs.

Offset:	0	1	2	3	4	5	6	7
Theta:	0.2957	0.0762	0.1014	0.0518	0.0518	0.1973	0.0762	0.1494

Table 5.15: Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized HPMs.

Offset:	0	1	2	3	4	5	6	7
Theta:	0.525	0.1	0.1	0.075	0.05	0.025	0.05	0.075

Table 5.16: Timing distribution for the Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions.

This chapter has provided the results of a variety of experiments on HPMs, using both the real sentence-picture experiment fMRI data and synthetic data. The next chapter discusses these results and summarizes our conclusions for Hidden Process Models.

Chapter 6

Conclusions

This thesis has presented Hidden Process Models, as inspired by the fMRI domain. We detailed the model formalism and inference and learning algorithms in Chapter 2. In Chapter 3, we described extensions to the standard version of HPMs, including regularization penalties for the process response signatures, a reparameterization of the process response signatures using basis functions, and a version of HPMs with continuous process start times. Chapter 4 discussed the theoretical correspondence between HPMs and Dynamic Bayes Nets, and gave suggestions for how the models could be learned in that framework. In Chapter 5 we provided experimental results evaluating the performance of HPMs on real and synthetic datasets, and we gave examples of how the learned models could be visualized.

In this chapter, we review the results presented in Chapter 5. We then suggest future directions for work on HPMs, provide some advice to potential users of HPMs, and revisit related work to illustrate the contributions of HPMs. Finally, we summarize and conclude.

6.1 Discussion of Results

The synthetic data results in Chapter 5 provide a proof-of-concept for the HPM approach. We showed that for data that is consistent with the assumptions of HPMs, HPMs can accurately recover the process response signatures. We also demonstrated that held-out data log-likelihood can be used to select the HPM that truly generated the data from among several HPMs with different numbers of processes. These results are important because they are obtained using data simulated to mimic the real fMRI experiment, but in a situation where we know the HPM modeling assumptions are correct and where we know

ground truth. Two key components of the thesis of this research were to recover the spatial-temporal signatures of processes with uncertain and variable onset, and to compare several models of an fMRI study. Since in the real data we do not know the true process response signatures or the true number of processes underlying the data, we provide the synthetic data results to demonstrate success at both of these tasks where we do have ground truth.

The results comparing models and methods using cross-validated data log-likelihood improvement over the baseline on real fMRI data indicated that HPMs trained using the standard parameterization and algorithm tend to overfit, as evidenced by log-likelihoods that are lower than the baseline. However, the HPMs using regularization or basis functions do not appear to overfit, consistently outperforming the baseline method. HPMs with regularization generally perform similarly to HPMs with basis functions. The improvement over standard HPMs that regularization and basis functions have in common is temporal smoothness constraints on the process response signatures, and our results suggest that these constraints are important.

Chapter 5 also provided cross-validated classification accuracies on the ViewPicture vs. ReadSentence task for the real fMRI experiment. These accuracies are helpful because they have a concrete interpretation, whereas the scale of the data log-likelihood scores is harder to interpret. However, they also measure performance on a task that only involves the first part of the trial, and aggregate spatial and temporal information (which may be interesting in its own right) into a binary decision. We saw that HPM-2-K outperformed HPM-GNB on this classification task, where the difference between those two models was that HPM-2-K allowed the ViewPicture and ReadSentence processes to have temporal overlap. Another trend in the classification results was that the HPMs containing more than just the two processes involved in the classification task did not perform as well as HPMs with just ViewPicture and ReadSentence. We suggest that this is because the extra processes being modeled in the more complex HPMs do not have much spatial-temporal overlap with the first two processes. If there were more spatial-temporal overlap, modeling the Decide and/or PressButton processes might aid classification, for example by deconvolving the beginning of the third process from the end of the second. However in the absence of signification spatial-temporal overlap, the additional parameters required to model the extra processes may force the algorithms to make compromises at the expense of the ViewPicture and ReadSentence process, degrading performance.

In general, our results show that HPMs do not outperform a Gaussian Naive Bayes classifier on this task, but we note that GNB has some advantages over HPMs for classification, including the ability to model class-conditional variances over time and space, as opposed to the process-independent, time-independent, voxel-wise variances in HPMs. We note that it would not be difficult to extend HPMs to model a variance for each point

in time and space over the trials, as long as all trials are the same length. We could also imagine an extension to the generative model that includes variances that depend on the processes, but this would require design choices about how to combine variances when process instances overlap, and how to model variance when no process instances are active. Process-specific variances may also introduce additional challenges for learning the model. While HPM accuracies were often lower than GNB accuracies, they were usually well above naive guessing, indicating that while HPMs are not optimized for classification, they can still recover some information helpful for discriminating among proceses.

In addition to estimating HPM performance with cross-validation, we have demonstrated how trained HPMs can be visualized by modelers, which can be helpful for exploratory data analysis. We must reiterate though, that it is critical to consider the assumptions of the model carefully when browsing learned HPMs.

6.2 Advice on Using HPMs

When considering applying HPMs to an fMRI study, the goal of the analysis should be kept in mind. If the task of interest is to classify among a set of stimuli based on the fMRI data in a fixed-length window beginning at the stimulus presentation, HPMs may not be the right tool. There are numerous other classification techniques that are optimized more specifically for finding decision boundaries, and as we have demonstrated, HPMs do not outperform Gaussian Naive Bayes on the ReadSentence vs. ViewPicture classification task. However, if the main interest in the study involves a cognitive process whose onset is not perfectly tied to the stimulus timing, HPMs may be quite relevant.

Also of note is that the trial structure of the sentence-picture fMRI data provides important conditional independencies that allow for more compact representation of configurations. Since trials are long enough to allow brain activity to return to a baseline after the mental task, there are timepoints twoard the end of each trial where no processes are active. This allows configurations to be specified at the trial-level, independently of the set of configurations for the neighboring trials; that is, the trial-specific latent indicator variables are conditionally indpendent given the process parameters. If trials were not separated by these inactive periods, we would have to consider configurations spanning the entire experiment instead of individual trials, which would likely consist of all possible combinations of the trial-specific configurations and be computationally intractable. For fMRI analysts, this means that HPMs are better suited to event-related designs (like the sentence-picture experiment) than block designs (in which many stimuli of the same type are presented in quick succession for a period of time). In other domains, HPM users are advised to exploit

similar structure in the time series of interest to avoid excess complexity.

Finally, when using HPMs it is wise to consider whether the processes of interest are identifiable. Processes are identifiable if there is sufficient variability in their timing over the course of the experiment to permit a unique solution to their deconvolution. Alternatively, processes are unidentifiable if there are multiple settings of their parameters that produce the same log-likelihood of the data. For instance, an obviously problematic case is when two processes of interest always overlap completely. Unfortunately, most cases are not as intuitive. The two-process HPMs in the sentence-picture experiment are identifiable because of the stimuli are presented in both possible orders. However, when the third processes is added, the situation is less clear. If the timing of the third process is always the same, then the end of the ViewPicture and ReadSentence response signatures and the beginning of the Decide response signature become unidentifiable. If the timing of the third process varies sufficiently across trials, we may have enough constraints to render the processes identifiable. In general, we can improve identifiability by varying the likely orders and timings of the processes of interest in the experiment.

Another way to guarantee identifiability is to experimentally isolate each process of interest. For example, we could augment the sentence-picture experiment with several preceding trials of single stimuli. However, this approach becomes less attractive when we consider the linearity assumption. While assuming that overlapping hemodynamic responses sum linearly is common and somewhat reasonable, deviations from linearity are also well-studied. The ViewPicture process by itself may be significantly different from the ViewPicture process in the context of the mental task at hand. For this reason, we recommend varying the context, order, and timing of the processes instead of isolating them to achieve identifiability.

6.3 Contributions

In section 1.3, we overviewed research relevant to Hidden Process Models. As discussed there, most efforts to estimate hemodynamic responses assume that the timings of their generating cognitive processes are known. Classification techniques also make this assumption. HPMs provide a framework for fMRI analysis that relaxes the assumption of fixed, known process timings, and allows both classification and HRF estimation.

One technique for HRF estimation that we discussed above is the General Linear Model (Dale and Buckner [1997], Dale [1999]). HPMs can be viewed as a generalization of the GLM. Instead of using a fixed design matrix containing the known experimental design, HPMs use a design matrix containing several competing explanations for some

time points. The EM algorithm discussed in Chapter 2 then iterates between determining which explanation is correct, and estimating the parameters of the model.

Above, we also mentioned SPM (Friston [2003]) as a widely used fMRI analysis technique, and we return to it here to clarify the difference between that approach and HPMs. In particular, SPM generates maps that indicate the extent to which particular voxels have activation that is statistically significant for various experimental conditions. The HPM maps in Chapter 5 show the learned parameters for the signatures of various processes. They do not make claims about statistical significance. We believe these maps are still helpful, but they should not be interpreted in the same way that SPM maps are. In general, SPM is a massively univariate technique that assumes known process timing to make statements about the statistical significance of voxel activity in various conditions. In contrast, HPMs are a multivariate technique that learns parameters of processes with uncertain timing.

Overall, we believe that the biggest impact of HPMs is to provide a framework for studying cognitive processes with unknown onset. Conveniently, HPMs also allow classification, although as we have discussed, they are not optimized for that task. Our work on regularizing HPMs and parameterizing HPMs with basis functions is a starting point for providing a variety of modeling assumptions about the process response signatures of interest, to accommodate researchers who may wish to model the hemodynamic response function in different ways.

6.4 Future Directions

We see four major avenues for future work on Hidden Process Models: work on the representation of the hypothesis space, work on different parameterizations of HPMs, work on automatically discovering the number of processes in an HPM, and work applying HPMs to more studies.

One current limitation of HPMs is the need to enumerate the hypothesis space of the model in the set of configurations. This is inelegant at best, and intractable at worst. Chapter 4 outlines a possible alternative to configurations using Dynamic Bayesian Networks. It remains to be seen whether the variational EM and/or Gibbs sampling algorithms for HPMs described in Chapter 4 can cope with data as high-dimensional, sparse, and noisy as fMRI. Further work on these approaches would be valuable since it could save work for HPM users (by not needing to specify the list of configurations for each trial) and allow the learned models to be independent of the sets of configurations.

A second direction for improving HPMs is to expand the choices for parameterizing the model. For instance, it would be interesting to introduce a parameter for the duration of each process, to be learned instead of fixed in advance. Experimenting with alternative noise models may also be fruitful, since we have seen that the Gaussian Naive Bayes noise model outperforms similar HPMs in classification.

Thirdly, it would be very interesting to develop algorithms for automatically discovering the optimal number of processes in an HPM. The current method of using cross-validated data log-likelihood to decide the correct number of processes is reasonable for choosing among a small set of possible numbers of processes, but becomes impractical when there is more uncertainty about how many processes to model.

Finally, there is still work to be done using the HPM framework developed so far. HPMs could be applied to more datasets, within the fMRI domain and outside of it. For scenarios with appropriate prior knowledge, it would be interesting to experiment with more regularization penalties, including spatial sparsity and spatial priors. Since HPMs parameterized with basis functions have performed well so far, it may be worth investigating different numbers of basis functions, different types of bases, and regularization for spatial smoothness on the basis function weights.

6.5 Summary

In conclusion, recall the thesis presented in Section 1.4: The thesis of the research reported here is that we can develop machine learning techniques for time series data that can simultaneously estimate parameters of temporal signatures and the onsets of the latent processes that generate them. These techniques can be improved by incorporating domain knowledge, and they can be used to express, learn, and compare competing models of the processes generating the data. This thesis is motivated by, and will be explored within the fMRI domain.

In support of this thesis, we introduced Hidden Process Models, with the formalism and algorithms in Chapter 2. We elaborated on this basic version of HPMs in Chapters 3 and 4. Then in Chapter 5, we provided results on synthetic fMRI data (for which we knew ground truth) demonstrating that HPMs can recover the parameters of the processes even in the presence of uncertain timing. We also used synthetic data to show that HPMs can use held-out data log-likelihood to select the model with the correct number of latent processes. We showed that incorporating domain knowledge about the temporal smoothness of the process response signatures in fMRI, via regularization and basis functions, improved the performance of HPMs on real fMRI data. We also provided a comparison

of several models of the real fMRI study, and we showed how the learned HPMs can be visualized to aid interpretation of the processes. Finally, the current chapter has discussed the experimental results, advice for HPM users, contributions of HPMs to the field, and future directions of research.

In summary, Hidden Process Models are, to the best of our knowledge, the first attempt to simultaneously recover the timings and hemodynamic responses of the cognitive processes underlying an fMRI experiment. The evidence to date suggests that HPMs have promise for fMRI, but they are still in their infancy. We hope that progress on the research directions outlined above will make HPMs increasingly useful to cognitive scientists.

Appendix A

Notation

This section provides tables of notation to aid the reader in understanding the formalism and algorithms for HPMs (Sections 2.1 and 2.2, respectively). Table A.1 refers primarily to the notation for the formalism, and Table A.2 is primarily for the algorithms.

Symbol	Meaning	
$\mathbf{Y}$	Data matrix.	$T \times V$
t	Time point in data.	$\{1 \ldots T\}$
v	Voxel.	$\{1 \ldots V\}$
π	Process.	$\{1 \ldots \lvert\Pi\rvert\}$
$d(\pi)$	The duration of process π.	
$\mathbf{W}(\pi)$	The process response signature matrix for process π.	$d(\pi) \times V$
$\Theta(\pi)$	Vector of multinomial parameters over $\Omega(\pi)$.	Same length as $\Omega(\pi)$
$\Omega(\pi)$	The vector of offsets for process π.	Integers
Π	The set of all processes.	
i	Process instance.	$\{1 \ldots I\}$
$\pi(i)$	The process ID for i.	
$\lambda(i)$	The timing landmark for i.	
$O(i)$	The offset for i.	
c	Configuration.	$\{1 \ldots C\}$
$\mathcal{C}$	The set of all configurations.	
σ_v^2	The variance of voxel v.	
$\delta(\cdot)$	Returns 1 if and only if its argument is true.	
$\mu(\mathbf{c})$	The mean of the distribution over $\mathbf{Y}$ under c.	$T \times V$
$\mathbf{X_c}$	The binary design matrix for configuration c.	$T \times D$
D	The sum of the durations of all processes ($\sum_\pi d(\pi)$).	
$\mathbf{W}$	The vertical concatenation of $\mathbf{W}(\pi) \forall \pi$.	$D \times V$
$\mu(c)$	The mean of the Gaussian distribution over Y under c.	$TV \times 1$
n	Trial.	$\{1 \ldots N\}$
c_n	Configuration within trial n.	$\{1 \ldots C_n\}$
$\mathcal{C}_n$	The set of all configurations for trial n.	

Table A.1: Notation used in the formalism for HPMs, roughly in order of introduction. The last column indicates the size and/or domain of the variable, where appropriate.

Symbol	Meaning	
Ψ	The set of HPM parameters to be learned from data.	
$\tilde{\mathbf{X}}_{\mathbf{c}}$	The vertical concatenation of $\tilde{\mathbf{X}}_{\mathbf{nc}}$ for all trials n.	$VT \times VD$
Y	Data vector.	$TV \times 1$
Y_n	Data vector for a trial n.	$T_nV \times 1$
W	The vertical concatenation of $W(\pi) \forall \pi$.	$DV \times 1$
σ^2	The vertical concatenation of σ_n^2 for all trials.	$VT \times 1$
Σ^2	The covariance matrix. $\Sigma = diag(\sigma^2)$	$VT \times VT$
Z	Binary indicator vector for the correct configuration.	$C \times 1$, $\sum_c Z_c = 1$
Z_c	Binary indicator variable for configuration c.	$Z_c \in \{0, 1\}$
M	Length of the vertical concatenation of the design matrices for all configurations in all trials. $M = V \sum_n T_n C_n$	
$\tilde{\mathbf{X}}$	The vertical concatenation of $\tilde{\mathbf{X}}_{\mathbf{n}}$ for all trials n.	$M \times VD$
$\tilde{\mathbf{X}}_{\mathbf{n}}$	The vertical concatenation of $\tilde{\mathbf{X}}_{\mathbf{c}}$ for all configurations in trial n.	$VT_nC_n \times VD$
$\tilde{\mathbf{X}}_{\mathbf{nc}}$	The block-diagonal of $\tilde{\mathbf{X}}_{\mathbf{nc}}$ is $\mathbf{X}_{\mathbf{nc}}$ repeated V times.	$VT_n \times VD$
$\tilde{Y}$	Vertical concatenation $\tilde{Y}_n$ for all n.	$M \times 1$
$\tilde{Y}_n$	Vertical concatenation of Y_n repeated for C_n times.	$VT_nC_n \times 1$
$\tilde{Z}$	Expansion of Z. Z_{nc} for each element m corresponding to c_n.	$M \times 1$
$\tilde{\mathbf{Z}}$	$\tilde{\mathbf{Z}} = diag(\tilde{Z})$	
$\tilde{\mathbf{\Sigma}}$	Covariance matrix for $\tilde{Y}$.	$M \times M$
$\tilde{\sigma^2}$	Diagonal of $\tilde{\mathbf{\Sigma}}$. σ_v^2 for each element m corresponding to voxel v.	$M \times 1$
$\mathbf{\Upsilon}$	Weights for the weighted least squares solution. $\mathbf{\Upsilon} = \tilde{\mathbf{\Sigma}}^{-1} E[\tilde{\mathbf{Z}}]$	
Z_{nc}	Binary indicator variable for configuration c in trial n.	$Z_{nc} \in \{0, 1\}$

Table A.2: Notation used in descriptions of the HPM algorithms. The last column indicates the size and/or domain of the variable, where appropriate.

Appendix B

Derivation of the M step Update for W

The M step estimate of $\mathbf{W}$ under uncertainty about the true configuration of process instances can be understood in terms of an extension to the General Linear Model (GLM) presented in 2.2.2. In that approach, we created a design matrix $\tilde{\mathbf{X}}_\mathbf{c}$ from the known configuration c. Here, we consider the design matrix $\tilde{\mathbf{X}}$ reflecting all possible configurations. The construction of $\tilde{\mathbf{X}}$ and the related variables $\tilde{Y}$, $\tilde{Z}$, and $\tilde{\mathbf{\Sigma}}$ the are described in Section 2.2.3.

We begin with the expected log-likelihood of the complete data, including the observed data $\mathbf{Y}$ and the latent indicator variable Z:

$$\begin{aligned}
Q(\Psi, \Psi^{\text{old}}) &= E_{P(Z|\mathbf{Y},\Psi^{\text{old}})}[\ln P(\mathbf{Y}, Z|\Psi)] \\
&= E_{P(Z|\mathbf{Y},\Psi^{\text{old}})}[\ln P(\mathbf{Y}|Z,\Psi) + \ln P(Z|\Psi)] \\
&= E_{P(Z|\mathbf{Y},\Psi^{\text{old}})}\left[\ln\left(\frac{1}{(2\pi)^{\frac{M}{2}}|\tilde{\mathbf{\Sigma}}|^{\frac{1}{2}}} exp\left(-\frac{1}{2}\tilde{Z}||\tilde{Y} - \tilde{\mathbf{X}}W||^2_{\tilde{\sigma}^{-1}}\right)\right)\right] + \\
&\quad E_{P(Z|\mathbf{Y},\Psi^{\text{old}})}\left[\ln\left(\prod_n^N \prod_{c\in\mathcal{C}_n} P(c|\Psi)^{Z_{nc}}\right)\right] \\
&= -\ln((2\pi)^{\frac{M}{2}}|\tilde{\mathbf{\Sigma}}|^{\frac{1}{2}}) - \frac{1}{2}E[\tilde{Z}]||\tilde{Y} - \tilde{\mathbf{X}}W||^2_{\tilde{\sigma}^{-1}} + \\
&\quad \sum_n^N \sum_{c\in\mathcal{C}_n} E[Z_{nc}]\ln(P(c|\Psi))
\end{aligned} \tag{B.1}$$

We can maximize the expected log-likelihood of the complete data Q by solving for

the W that minimizes the objective function:

$$\begin{aligned} f_0(W) &= E[\tilde{Z}]||\tilde{Y} - \tilde{\mathbf{X}}W||^2_{\tilde{\sigma}^{-1}} \\ &= (\tilde{Y} - \tilde{\mathbf{X}}W)'\tilde{\mathbf{\Sigma}}^{-1}E[\tilde{\mathbf{Z}}](\tilde{Y} - \tilde{\mathbf{X}}W) \end{aligned} \tag{B.2}$$

This is a weighted least squares problem with weight matrix $\mathbf{\Upsilon} = \tilde{\mathbf{\Sigma}}^{-1}E[\tilde{\mathbf{Z}}]$, the solution to which is given by:

$$\hat{W} \longleftarrow \left(\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} \tag{B.3}$$

We arrive at this solution by setting the derivative of $f_0(W)$ equal to zero and solving for W as follows:

$$\begin{aligned} \frac{df_0}{dW} &= \frac{d}{dW}(\tilde{Y} - \tilde{\mathbf{X}}W)'\mathbf{\Upsilon}(\tilde{Y} - \tilde{\mathbf{X}}W) \\ &= \frac{d}{dW}\left(\tilde{Y}'\mathbf{\Upsilon}\tilde{Y} - 2\tilde{Y}'\mathbf{\Upsilon}\tilde{\mathbf{X}}W + W'\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}W\right) \\ 0 &= -2\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} + 2\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}W \\ \tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}W &= \tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} \\ W &= \left(\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}'\mathbf{\Upsilon}\tilde{Y} \end{aligned} \tag{B.4}$$

Appendix C

Tutorial on Factorial Hidden Markov Models

In this section, we give some background on factorial Hidden Markov Models (fHMMs) which may be helpful in understanding the extension to fHMMs that corresponds to Hidden Process Models, described in Chapter 4. We describe the model, two exact inference algorithms, and two variational inference algorithms for fHMMs. None of this is original work; instead, it is intented as a tutorial for the unfamiliar reader. All of the material on fHMMs follows Ghahramani and Jordan [1997] closely. The only original work in this Appendix is in Section C.1.3, which describes a small modification to the second fHMM exact inference algorithm for HPMs.

Factorial Hidden Markov Models are a generalization of Hidden Markov Models (HMMs) (Rabiner [1989]) in which the hidden state is factored into multiple state variables that are conditionally independent of one another given the observed data. A graphical depiction of the model is given in Figure C.1. Let $\{S_t\}$ be a sequence of hidden states, using the factored representation with M chains so that $S_t = \{S_t^{(1)}, S_t^{(2)}, \ldots, S_t^{(M)}\}$, where $S_t^{(m)}$ is a $K \times 1$ binary vector containing a single 1 in the position of the state value. In general each chain can take on $K^{(m)}$ values but here for simplicity, we assume that $K^{(m)} = K, \forall m$. Let $\{Y_t\}$ be a sequence of observations, and let Ψ represent the parameters of the model. The factorial HMM defines a probability distribution as follows:

$$P(\{S_t, Y_t\}|\Psi) = \prod_{m=1}^{M} \left[P(S_1^{(m)}|\Psi) \right] P(Y_1|S_1, \Psi) \prod_{t=2}^{T} \left[\prod_{m=1}^{M} \left[(P(S_t^{(m)}|S_{t-1}^{(m)}, \Psi) \right] P(Y_t|S_t, \Psi) \right] \tag{C.1}$$

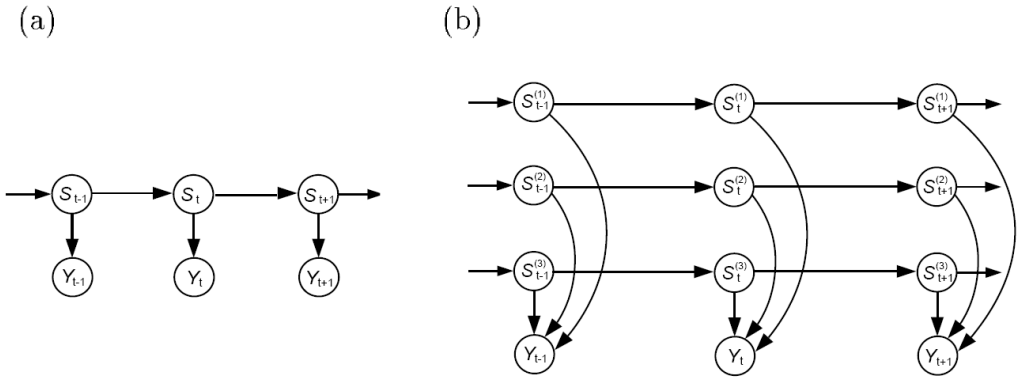

Figure C.1: (a) Graphical representation of a Hidden Markov Model. S_t is the value of the hidden state at time t; Y_t is the observed data at time t. (b) Graphical representation of a factorial Hidden Markov Model with $M = 3$ hidden Markov chains. $S_t^{(m)}$ is the value of the hidden state of the m^{th} chain at time t. This figure is from Ghahramani and Jordan [1997].

We will work with the following distributions for the components of $P(\{S_t, Y_t\}|\Psi)$:

$$P(S_1^{(m)}|\Psi) = \prod_{k=1}^{K} \left(\pi_k^{(m)}\right)^{S_{1,k}^{(m)}} \tag{C.2}$$

$$\log P(S_1^{(m)}|\Psi) = S_1^{(m)} \log(\pi^{(m)}) \tag{C.3}$$

where $\pi^{(m)}$ is a $K \times 1$ vector containing the initial probabilites of each state value for chain m.

$$P(S_t^{(m)}|S_{t-1}^{(m)}, \Psi) = \prod_{k=1}^{K} \left(\prod_{j=1}^{K} \left(A_{j,k}^{(m)}\right)^{S_{t-1,j}^{(m)}} \right)^{S_{t,k}^{(m)}} \tag{C.4}$$

$$\log P(S_t^{(m)}|S_{t-1}^{(m)}, \Psi) = (S_{t-1}^{(m)})' \log(A^{(m)}) S_t^{(m)} \tag{C.5}$$

where $A^{(m)}$ is a $K \times K$ matrix containing the state transition probabilities for chain m

($A_{i,j}^{(m)}$ is $P(S_t^{(m)} = j | S_{t-1}^{(m)} = i)$).

$$
\begin{aligned}
P(Y_t|S_t,\Psi) &= |C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}} \times \\
&\quad \exp\left(-\frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1} (Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})\right) \quad \text{(C.6)} \\
\log P(Y_t|S_t,\Psi) &= \log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) - \\
&\quad \frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1} (Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)}) \quad \text{(C.7)} \\
& \quad \text{(C.8)}
\end{aligned}
$$

where Y_t is a $D \times 1$ vector of observations, C is a $D \times D$ covariance matrix, and $W^{(m)}$ is a $D \times K$ matrix where $W_{i,j}^{(m)}$ is the contribution to the i^{th} dimension of Y_t from chain m when it takes on its j^{th} value. This equation says that the observed data is governed by a multivariate Gaussian distribution whose mean is the sum of a contribution from each chain that depends on the chain's value. Using these distributions, $\Psi = \{\{\pi^{(m)}, A^{(m)}, W^{(m)}\} \forall m, C\}$.

For the inference problem, we are interested in the probability distribution over the hidden state sequence conditioned on the observation sequence, which is:

$$
\begin{aligned}
P(\{S_t\}|\{Y_t\},\Psi) &= \frac{P(\{S_t, Y_t\}|\Psi)}{P(\{Y_t\}|\Psi)} \\
&= \frac{1}{Z_P} P(\{S_t, Y_t\}|\Psi) \\
&= \frac{1}{Z_P} \prod_{m=1}^{M} \left[P(S_1^{(m)}|\Psi)\right] P(Y_1|S_1,\Psi) \prod_{t=2}^{T} \left[\prod_{m=1}^{M} \left[(P(S_t^{(m)}|S_{t-1}^{(m)},\Psi)\right] P(Y_t|S_t,\Psi)\right] \quad \text{(C.9)}
\end{aligned}
$$

$$
\begin{aligned}
\log P(\{S_t\}|\{Y_t\},\Psi) &= -\log(Z_P) + \sum_{m=1}^{M} \log P(S_1^{(m)}|\Psi) + \sum_{t=1}^{T} \log P(Y_t|S_t,\Psi) + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T} \log P(S_t^{(m)}|S_{t-1}^{(m)},\Psi) \\
&= -\log(Z_P) + \sum_{m=1}^{M} S_1^{(m)} \log(\pi^{(m)}) + \sum_{m=1}^{M}\sum_{t=2}^{T} (S_{t-1}^{(m)})' \log(A^{(m)}) S_t^{(m)} + \\
&\quad \log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) - \frac{1}{2}\sum_{t=1}^{T} Y_t' C^{-1} Y_t + \sum_{t=1}^{T}\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} S_t^{(m)} - \\
&\quad \frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{n=1}^{M} (S_t^{(n)})'(W^{(n)})' W^{(m)} S_t^{(m)}
\end{aligned}
\tag{C.10}
$$

C.1 Exact Inference

C.1.1 Naive Inference

Since a factorial HMM can be expressed as an HMM, we can use the forward-backward algorithm (Rabiner [1989]) to do inference in fHMMs. For a fHMM with M state variables, each of which takes on K values, the corresponding HMM has K^M states. The forward-backward algorithm proceeds by computing $\alpha_t(i) = P(Y_{1:t}, S_t = i|\Psi)$ for every time t and state value i on the forward pass, and $\beta_t(i) = P(Y_{t+1:T}|S_t = i, \Psi)$ for every time t and state value i on the backward pass, where $Y_{a:b}$ indicates the sequence from a to b. The forward-backward algorithm can be implemented with dynamic programming. The forward recursion is given by:

$$
\begin{aligned}
\alpha_1(i) &= P(S_1 = i)P(Y_1|S_1 = i) \\
&\quad 1 \le i \le K^M
\end{aligned}
\tag{C.11}
$$

$$
\begin{aligned}
\alpha_{t+1}(j) &= P(Y_{t+1}|S_{t+1} = j)\sum_{i=1}^{K^M} P(S_{t+1} = j|S_t = i)\alpha_t(i) \\
&\quad 1 \le t \le T-1, 1 \le j \le K^M
\end{aligned}
\tag{C.12}
$$

The backward recursion is given by:

$$\begin{aligned} \beta_T(i) &= 1 \\ & 1 \leq i \leq K^M \end{aligned} \tag{C.13}$$

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^{K^M} P(Y_{t+1}|S_{t+1}=j)P(S_{t+1}=j|S_t=i)\beta_{t+1}(j) \\ & t=[T-1,T-2,\ldots,1], 1 \leq i \leq K^M \end{aligned} \tag{C.14}$$

The complexity of the forward-backward algorithm is $O(TK^{2M})$. The TK^M in the complexity result occurs because we do a computation for every time step and every state value, and the second K^M occurs because the sum over $P(S_{t+1}=j|S_t=i)$ also has K^M terms.

C.1.2 An Improvement

We can improve on the naive inference algorithm presented above by taking advantage of the conditional independence assumption between Markov chains in fHMMs. In this case, the transition probability matrix can be factored into M $K \times K$ matrices, one for each state variable M, instead of the $K^M \times K^M$ matrix used above. For the forward recursion, this gives:

$$\sum_{i=1}^{K^M} P(S_{t+1}=j|S_t=i) = \alpha_t(i) \prod_{m=1}^{M} \sum_{i^{(m)}}^{K^{(m)}} P(S_{t+1}^{(m)}=j|S_t^{(m)}=i^{(m)}) \tag{C.15}$$

There are K terms in each sum, and M sums overall, resulting in KM terms instead of K^M. This improves our complexity result from $O(TK^{2M})$ to $O(TKMK^M) = O(TMK^{M+1})$.

C.1.3 Exact Inference for HPMs

Here, we attempt to modify the exact inference algorithms presented above for HPMs. Our hope is that the additional structure provided by the HPM assumptions will improve the complexity of the algorithms.

The first improvement we can make on the fHMM inference algorithm using the HPM structure is to remove the $+1$ from the time complexity $O(TMK^{M+1})$. We do this by

noticing that for any given state S_t, the sum over probabilities of transitioning from any of the K^M states to S_t can be simplified. If $S_t = 0$, then S_{t-1} could only have been 0 or d, so the sum is $P(S_t = 0|S_{t-1} = 0) + P(S_t = 0|S_{t-1} = d) = (1 - p) + 1$. If $S_t > 0$, the state variable is deterministically counting, so the only non-zero term in the sum over all possible predecessors is $P(S_t = x|S_{t-1} = x - 1) = 1$. So we have:

$$\begin{aligned}\prod_{m=1}^{M}\sum_{i}^{K} P(S_{t+1}^{(m)} = j^{(m)}|S_t^{(m)} = i) &= (1-p)+1, j^{(m)} = 0 \\ &= 1, j^{(m)} > 0 \end{aligned} \tag{C.16}$$

Decomposing this sum decouples it from the arity of the state variable, K, which means a constant amount of computation for each state variable, so we have $O(TMK^M)$.

The second insight for HPMs as fHMMs is that if all the state variables $j^{(m)}$ for for state value j are non-zero, then we have:

$$\prod_{m=1}^{M}\sum_{i}^{K} P(S_{t+1}^{(m)} = j^{(m)}|S_t^{(m)} = i) = \prod_{m=1}^{M} 1 = 1 \tag{C.17}$$

For these state values, the forward recursion simplifies to:

$$\alpha_{t+1}(j) = P(Y_{t+1}|S_{t+1} = j)\prod_{m=1}^{M}\alpha_t(j^{(m)} - 1) \tag{C.18}$$

Finally, note that there are $2^M - 1$ state values with at least one zero for one of the state variable values, regardless of the arity of the state variables.

Unfortunately this second insight does not further improve our complexity, since even though we can remove the sum completely, we still have a product of M terms for each of K^M state variables.

C.2 Variational Inference

Since the exact inference algorithms presented so far all have exponential time complexities, we now consider approximate inference algorithms for fHMMs. We will begin by explaining the theory behind variational inference in general, and then give two examples of variational inference algorithms for fHMMs.

C.2.1 The General Framework for Variational Inference

Here we describe the basic idea behind variational approximations for inference in graphical models. This section is based on the descriptions given in Jordan et al. [1998] and Ghahramani and Jordan [1997].

Let us call the set of hidden variables in our model H and the set of observed variables, or evidence E. The inference problem is to compute the probability distribution over the hidden variables given the evidence:

$$\begin{aligned} P(H|E) &= \frac{P(H,E)}{P(E)} \\ &= \frac{P(H,E)}{\sum_H P(H,E)} \end{aligned} \tag{C.19}$$

When $P(H|E)$ is intractable to compute directly, we can use variational methods to approximate it by a tractable distribution $Q(H|E)$. Here we show that any distribution $Q(H|E)$ provides a lower bound on the log likelihood of $P(E)$:

$$\begin{aligned} \log P(E) &= \log \sum_H P(H,E) \\ &= \log \sum_H Q(H|E) \frac{P(H,E)}{Q(H|E)} \\ &\geq \sum_H Q(H|E) \log \frac{P(H,E)}{Q(H|E)} \qquad \text{(C.20)} \\ &= \sum_H Q(H|E) \log \frac{P(H|E)P(E)}{Q(H|E)} \\ &= \sum_H Q(H|E) (\log \frac{P(H|E)}{Q(H|E)} + \log P(E)) \\ &= \sum_H Q(H|E) \log \frac{P(H|E)}{Q(H|E)} + \sum_H Q(H|E) \log P(E) \\ &= \sum_H Q(H|E) \log \frac{P(H|E)}{Q(H|E)} + \log P(E) \qquad \text{(C.21)} \\ & \qquad \text{(C.22)} \end{aligned}$$

where in Equation C.20 we have used Jensen's inequality (Wasserman [2004]). Subtracting $\log P(E)$ from both sides, we see that the difference between the left and right sides

is:

$$\begin{aligned} 0 &\geq \sum_H Q(H|E) \log \frac{P(H|E)}{Q(H|E)} \\ 0 &\leq \sum_H Q(H|E) \log \frac{Q(H|E)}{P(H|E)} \end{aligned} \tag{C.23}$$

This last line is the Kullback-Leibler (KL) divergence (Wasserman [2004]) between Q and P, $KL(Q||P)$. This means that the KL divergence between an arbitrary distribution $Q(H|E)$ and the true distribution $P(H|E)$ is a lower bound on the log likelihood of the observed evidence.

Variational inference methods proceed from this result by choosing a family of distributions $Q(H|E)$ defined by a set of conditional independence relationships among the variables in H, and then choosing a member of that family, which corresponds to choosing particular settings for the parameters of the family Q. These parameters are called variational parameters. To make the bound on the log likelihood as tight as possible, the variational parameters are chosen to minimize the KL divergence between Q and P (e.g. by setting the derivative of Equation C.23 to zero and solving for the parameters of Q).

Having chosen and parameterized Q, we can approximate $P(H|E)$ with $Q(H|E)$ and/or compute expected sufficient statistics under $Q(H|E)$ to be used in learning model parameters in an EM algorithm.

C.2.2 Mean Field Approximation for Factorial HMMs

In this section, we describe the mean field approximation for factorial HMMs. Both the method and the notation are based on Ghahramani and Jordan [1997].

The first step in developing a variational method is to choose the family of distributions to be used for Q; that is, to choose the set of conditional independence relationships that hold for Q. The simplest choice of Q for fHMMs is to decouple all of the state variables, breaking the links of each chain going forward in time and the links from each chain to the observed variables at every time point. This completely factorized approximation is called the mean field approximation, and is depicted in Figure C.2.

The family of distributions represented by this structure is given by:

$$Q(\{S_t\}|\theta) = \prod_{t=1}^{T} \prod_{m=1}^{M} Q(S_t^{(m)}|\theta_t^{(m)}) \tag{C.24}$$

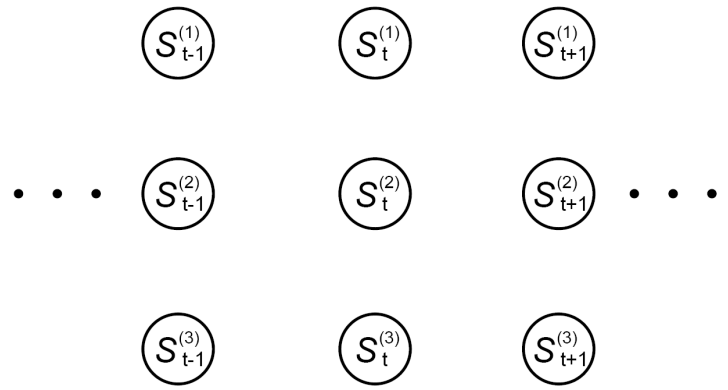

Figure C.2: Graphical representation of the approximating family of distributions for the mean field variational approximation for factorial HMMs. $S_t^{(m)}$ represents the state of the m^{th} chain at time t. In the mean field approximation, all state variables are decoupled. This figure is from Ghahramani and Jordan [1997].

where $\theta = \{\theta_t^{(m)}\}$ are the variational parameters, also represented as $K \times 1$ vectors defining a multinomial distribution on $S_t^{(m)}$, in which the k^{th} element contains the probability that $S_t^{(m)}$ is in state k:

$$Q(S_t^{(m)}|\theta_t^{(m)}) = \prod_{k=1}^{K} (\theta_{t,k}^{(m)})^{S_{t,k}^{(m)}} \tag{C.25}$$

Adding a normalization term Z_Q to ensure that $\sum_k \theta_{t,k}^{(m)} = 1$, we have:

$$\begin{aligned} Q(\{S_t\}|\theta) &= \frac{1}{Z_Q} \prod_{t=1}^{T} \prod_{m=1}^{M} \prod_{k=1}^{K} (\theta_{t,k}^{(m)})^{S_{t,k}^{(m)}} \\ \log Q(\{S_t\}|\theta) &= -\log(Z_Q) + \sum_{t=1}^{T} \sum_{m=1}^{M} (S_t^{(m)})' \log \theta_t^{(m)} \end{aligned} \tag{C.26}$$

Given this family of distributions, we wish to choose the variational parameters θ that minimize the KL-divergence between $Q(\{S_t\}|\theta)$ and the true $P(\{S_t\}|\{Y_t\}, \Psi)$, given by:

$$KL(Q||P) = \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log \frac{Q(\{S_t\}|\theta)}{P(\{S_t\}|\{Y_t\}, \Psi)} \tag{C.27}$$

$$KL(Q||P) = \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log Q(\{S_t\}|\theta) - \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log P(\{S_t\}|\{Y_t\}, \Psi) \tag{C.28}$$

$$\begin{aligned} KL(Q||P) &= \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(-\log(Z_Q) + \sum_{t=1}^{T} \sum_{m=1}^{M} (S_t^{(m)})' \log \theta_t^{(m)} + \log(Z_P) \right) - \\ & \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(\sum_{m=1}^{M} S_1^{(m)} \log(\pi^{(m)}) + \sum_{m=1}^{M} \sum_{t=2}^{T} (S_{t-1}^{(m)})' \log(A^{(m)}) S_t^{(m)} \right) - \\ & \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(\log(|C|^{\frac{1}{2}} (2\pi)^{\frac{D}{2}}) - \frac{1}{2} \sum_{t=1}^{T} Y_t' C^{-1} Y_t \right) - \\ & \sum_{\{S_t\}} Q(\{S_t\}|\theta) \sum_{t=1}^{T} \sum_{m=1}^{M} Y_t' C^{-1} W^{(m)}) S_t^{(m)} + \\ & \sum_{\{S_t\}} Q(\{S_t\}|\theta) \frac{1}{2} \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{n=1}^{M} (S_t^{(n)})' (W^{(n)})' W^{(m)} S_t^{(m)} \end{aligned} \tag{C.29}$$

$$\begin{aligned} KL(Q||P) &= -\log(Z_Q) + \sum_{t=1}^{T} \sum_{m=1}^{M} (E_Q[S_t^{(m)}])' \log \theta_t^{(m)} + \log(Z_P) - \\ & \sum_{m=1}^{M} E_Q[S_1^{(m)}] \log(\pi^{(m)}) - \sum_{m=1}^{M} \sum_{t=2}^{T} tr\{E_Q[S_t^{(m)} (S_{t-1}^{(m)})'] \log(A^{(m)})\} - \\ & \log(|C|^{\frac{1}{2}} (2\pi)^{\frac{D}{2}}) + \frac{1}{2} \sum_{t=1}^{T} Y_t' C^{-1} Y_t - \\ & \sum_{t=1}^{T} \sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} E_Q[S_t^{(m)}] + \\ & \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} tr\{diag\{E_Q[S_t^{(m)}]\} (W^{(m)})' C^{-1} W^{(m)}\} + \\ & \frac{1}{2} \sum_{m=1}^{M} \sum_{n \neq m} \sum_{t=1}^{T} tr\{E_Q[S_t^{(m)} (S_t^{(n)})'] (W^{(n)})' C^{-1} W^{(m)}\} \end{aligned} \tag{C.30}$$

where $tr\{\}$ is takes the trace of a matrix and $diag\{\}$ puts its argument on the diagonal of an otherwise-zero matrix.

In this case, the expected values of the state variables are simple because Q is fully factorized:

$$E_Q[S_t^{(m)}] = \theta_t^{(m)} \tag{C.31}$$

$$E_Q[S_t^{(m)}(S_{t-1}^{(m)})'] = \theta_t^{(m)}(\theta_{t-1}^{(m)})' \tag{C.32}$$

$$E_Q[S_t^{(m)}(S_t^{(n)})'] = \theta_t^{(m)}(\theta_t^{(n)})' \tag{C.33}$$

We now plug these expected values back into the KL divergence and take the derivative with respect to the variational parameters:

$$\begin{aligned}\frac{dKL(Q||P)}{d\theta_\tau^{(l)}} &= -\frac{d}{d\theta_\tau^{(l)}}\log(Z_Q) + 1 + \log\theta_\tau^{(l)} - (W^{(l)})'C^{-1}Y_\tau + \frac{1}{2}\Delta\{(W^{(l)})'C^{-1}W^{(l)}\} + \\ &\quad \sum_{n\neq l}^{M}(W^{(l)})'C^{-1}W^{(n)}\theta_t^{(n)} - \log A^{(m)}\theta_{\tau+1}^{(l)} - (\theta_{\tau-1}^{(l)})'\log A^{(l)}\end{aligned} \tag{C.34}$$

where $\Delta\{\}$ returns a vector containing the diagonal of its argument.

Setting this derivative to zero and solving for $\theta_\tau^{(l)}$ gives:

$$\begin{aligned}0 &= -\frac{d}{d\theta_\tau^{(l)}}\log(Z_Q) + 1 + \log\theta_\tau^{(l)} - (W^{(l)})'C^{-1}Y_\tau + \frac{1}{2}\Delta\{(W^{(l)})'C^{-1}W^{(l)}\} + \\ &\quad \sum_{n\neq l}^{M}(W^{(l)})'C^{-1}W^{(n)}\theta_t^{(n)} - \log A^{(m)}\theta_{\tau+1}^{(l)} - (\theta_{\tau-1}^{(l)})'\log A^{(l)}\end{aligned} \tag{C.35}$$

$$\begin{aligned}\log\theta_\tau^{(l)} &= \frac{d}{d\theta_\tau^{(l)}}\log(Z_Q) - 1 + (W^{(l)})'C^{-1}Y_\tau - \frac{1}{2}\Delta\{(W^{(l)})'C^{-1}W^{(l)}\} - \\ &\quad \sum_{n\neq l}^{M}(W^{(l)})'C^{-1}W^{(n)}\theta_t^{(n)} + \log A^{(m)}\theta_{\tau+1}^{(l)} + (\theta_{\tau-1}^{(l)})'\log A^{(l)}\end{aligned} \tag{C.36}$$

$$
\begin{aligned}
\hat{\theta}_\tau^{(l)} &= \exp(c + (W^{(l)})' C^{-1} Y_\tau - \frac{1}{2}\Delta\{(W^{(l)})' C^{-1} W^{(l)}\} - \\
&\quad \sum_{n \neq l}^{M} (W^{(l)})' C^{-1} W^{(n)} \theta_t^{(n)} + \log A^{(m)} \theta_{\tau+1}^{(l)} + (\theta_{\tau-1}^{(l)})' \log A^{(l)}) \\
&= \exp(c + (W^{(l)})' C^{-1} (Y_\tau - \sum_{n \neq l}^{M} (W^{(n)})' \theta_t^{(n)}) - \\
&\quad \frac{1}{2}\Delta\{(W^{(l)})' C^{-1} W^{(l)}\} + \log A^{(m)} \theta_{\tau+1}^{(l)} + (\theta_{\tau-1}^{(l)})' \log A^{(l)}) \\
&= \exp\left((W^{(l)})' C^{-1} \tilde{Y}_\tau^{(l)} - \frac{1}{2}\Delta\{(W^{(l)})' C^{-1} W^{(l)}\} + \log A^{(m)} \theta_{\tau+1}^{(l)} + (\theta_{\tau-1}^{(l)})' \log A^{(l)} \right)
\end{aligned}
\tag{C.37}
$$

where $\tilde{Y}_\tau^{(l)} = Y_\tau - \sum_{n \neq l}^{M} (W^{(n)})' \theta_t^{(n)}$, the residual error in Y_τ given the predictions from all the state variables except for l. The constant c has been removed, and in its place we renormalize the vector $\hat{\theta}_\tau^{(l)}$ after every computation.

Equation C.37 defines a set of fixed point equations for the variational parameters. We can iterate on these equations until the KL divergence converges. Notice that the terms in the fixed point equation for a particular $\theta_t^{(m)}$ involve the other parameters in the Markov blanket of $\theta_t^{(m)}$ (the variational parameters for chains $n \neq m$ at time t, and the variational parameters for chain m at times $t-1$ and $t+1$). Therefore, even though the approximating distribution is completely factorized, its parameters are coupled according to the dependencies in the true distribution.

Having solved for the variational parameters, we can use them to do inference about the hidden state sequence using Q or we can take the expectations under Q of the sufficient statistics of P to be used in an EM algorithm for learning the parameters Ψ of P.

C.2.3 Structured Approximation for Factorial HMMs

In this section, we describe a structured approximation for factorial HMMs. Again, both the method and the notation are based on Ghahramani and Jordan [1997].

As before, we start by choosing a family of distributions with which to approximated $P(\{S_t\}|\{Y_t\}, \Psi)$. This family is depicted in Figure C.3. Unlike the mean field approximation, this family of distributions retains the temporal structure of the hidden state variables. It essentially treats each chain as an independent HMM, decoupled from the observed data.

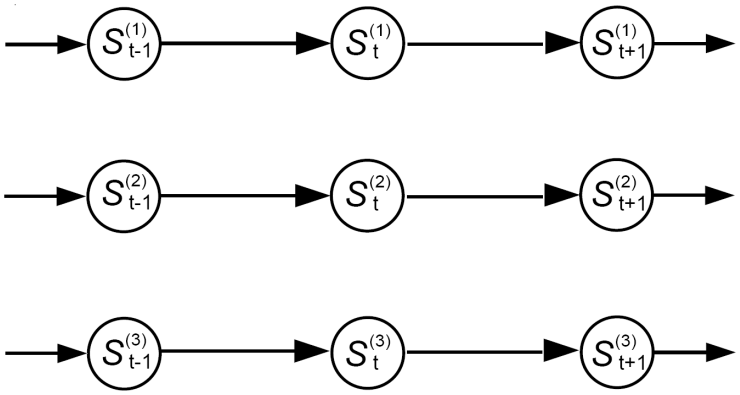

Figure C.3: Graphical representation of the approximating family of distributions for the structured variational approximation for factorial HMMs. $S_t^{(m)}$ represents the state of the m^{th} chain at time t. In this approximation, the temporal structure of each chain is retained, but the chains are treated independently (the arrows from the state variables to the observed variables are broken). This figure is from Ghahramani and Jordan [1997].

Let us write this family Q as:

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \prod_{m=1}^{M} Q(S_1^{(m)}|\theta) \prod_{t=2}^{T} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta) \tag{C.38}$$

We choose the following parameterization for the components of Q:

$$\begin{aligned} Q(S_1^{(m)}|\theta) &= \prod_{k=1}^{K} (h_{1,k}^{(m)} \pi_k^{(m)})^{S_{1,k}^{(m)}} \\ \log Q(S_1^{(m)}|\theta) &= \sum_{k=1}^{K} S_{1,k}^{(m)} \log(h_{1,k}^{(m)} \pi_k^{(m)}) \\ &= (S_1^{(m)})' \log h_1^{(m)} + (S_1^{(m)})' \log \pi^{(m)} \end{aligned} \tag{C.39}$$

$$\begin{aligned} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta) &= \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \prod_{j=1}^{K} (A_{j,k}^{(m)})^{S_{t-1,j}^{(m)}} \right)^{S_{t,k}^{(m)}} \\ \log Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta) &= (S_t^{(m)})' \log h_t^{(m)} + (S_{t-1}^{(m)})' \log A^{(m)} S_t^{(m)} \end{aligned} \tag{C.40}$$

The variational parameters are $\theta = \{\pi^{(m)}, A^{(m)}, h_t^{(m)}\} \forall m, t$, where $h_t^{(m)}$ is a $K \times 1$ vector that intuitively plays the role of the probability of a fictitious observation for each of the K settings of $S_t^{(m)}$. So we have:

$$\begin{aligned} Q(\{S_t\}|\theta) &= \frac{1}{Z_Q} \prod_{m=1}^{M} \prod_{k=1}^{K} (h_{1,k}^{(m)} \pi_k^{(m)})^{S_{1,k}^{(m)}} \prod_{t=2}^{T} \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \prod_{j=1}^{K} (A_{j,k}^{(m)})^{S_{t-1,j}^{(m)}} \right)^{S_{t,k}^{(m)}} \\ \log Q(\{S_t\}|\theta) &= -\log(Z_Q) + \sum_{m=1}^{M} (S_1^{(m)})' \log \pi^{(m)} + \sum_{m=1}^{M} \sum_{t=1}^{T} (S_t^{(m)})' \log h_t^{(m)} + \\ & \quad \sum_{m=1}^{M} \sum_{t=2}^{T} (S_{t-1}^{(m)})' \log A^{(m)} S_t^{(m)} \end{aligned} \tag{C.41}$$

We follow the same reasoning as above in determining how to set the parameters θ of Q; that is, we again seek to minimize the KL divergence between Q and P:

$$KL(Q||P) = \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log \frac{Q(\{S_t\}|\theta)}{P(\{S_t\}|\{Y_t\}, \Psi)} \tag{C.42}$$

$$KL(Q||P) = \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log Q(\{S_t\}|\theta) - \sum_{\{S_t\}} Q(\{S_t\}|\theta) \log P(\{S_t\}|\{Y_t\}, \Psi) \tag{C.43}$$

$$
\begin{aligned}
KL(Q||P) \;=\; & \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(-\log(Z_Q) + \sum_{m=1}^{M} (S_1^{(m)})' \log \pi^{(m)} + \sum_{m=1}^{M} \sum_{t=1}^{T} (S_t^{(m)})' \log h_t^{(m)} \right) + \\
& \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(\sum_{m=1}^{M} \sum_{t=2}^{T} (S_{t-1}^{(m)})' \log A^{(m)} S_t^{(m)} \right) - \\
& \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(-\log(Z_P) + \sum_{m=1}^{M} S_1^{(m)} \log(\pi^{(m)}) \right) - \\
& \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(\sum_{m=1}^{M} \sum_{t=2}^{T} (S_{t-1}^{(m)})' \log(A^{(m)}) S_t^{(m)} + \log(|C|^{\frac{1}{2}} (2\pi)^{\frac{D}{2}}) \right) - \\
& \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(-\frac{1}{2} \sum_{t=1}^{T} Y_t' C^{-1} Y_t + \sum_{t=1}^{T} \sum_{m=1}^{M} Y_t' C^{-1} W^{(m)}) S_t^{(m)} \right) - \\
& \sum_{\{S_t\}} Q(\{S_t\}|\theta) \left(-\frac{1}{2} \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{n=1}^{M} (S_t^{(n)})' (W^{(n)})' W^{(m)} S_t^{(m)} \right)
\end{aligned}
\tag{C.44}
$$

$$
\begin{aligned}
KL(Q||P) &= -\log(Z_Q) + \sum_{m=1}^{M}\sum_{t=1}^{T} E_Q[S_t^{(m)}]' \log h_t^{(m)} + \sum_{m=1}^{M} E_Q[S_1^{(m)}]' \log \pi^{(m)} + \\
&\sum_{m=1}^{M}\sum_{t=2}^{T} tr\{E_Q[S_t^{(m)}(S_{t-1}^{(m)})'] \log(A^{(m)})\} + \\
&\log(Z_P) - \sum_{m=1}^{M} E_Q[S_1^{(m)}] \log(\pi^{(m)}) - \\
&\sum_{m=1}^{M}\sum_{t=2}^{T} tr\{E_Q[S_t^{(m)}(S_{t-1}^{(m)})'] \log(A^{(m)})\} - \log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) + \\
&\frac{1}{2}\sum_{t=1}^{T} Y_t' C^{-1} Y_t - \sum_{t=1}^{T}\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} E_Q[S_t^{(m)}] + \\
&\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T} tr\{diag\{E_Q[S_t^{(m)}]\}(W^{(m)})' C^{-1} W^{(m)}\} + \\
&\frac{1}{2}\sum_{m=1}^{M}\sum_{n\neq m}\sum_{t=1}^{T} tr\{E_Q[S_t^{(m)}(S_t^{(n)})'](W^{(n)})' C^{-1} W^{(m)}\}
\end{aligned}
\tag{C.45}
$$

$$
\begin{aligned}
KL(Q||P) &= -\log(Z_Q) + \sum_{m=1}^{M}\sum_{t=1}^{T} E_Q[S_t^{(m)}]' \log h_t^{(m)} + \log(Z_P) + \\
&\frac{1}{2}\sum_{t=1}^{T} Y_t' C^{-1} Y_t - \sum_{t=1}^{T}\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} E_Q[S_t^{(m)}] + \\
&\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T} tr\{diag\{E_Q[S_t^{(m)}]\}(W^{(m)})' C^{-1} W^{(m)}\} + \\
&\frac{1}{2}\sum_{m=1}^{M}\sum_{n\neq m}\sum_{t=1}^{T} tr\{E_Q[S_t^{(m)}(S_t^{(n)})'](W^{(n)})' C^{-1} W^{(m)}\}
\end{aligned}
\tag{C.46}
$$

Note that we can write this as:

$$KL(Q||P) = -\log(Z_Q) + \log(Z_P) + \sum_{m=1}^{M}\sum_{t=1}^{T} E_Q[S_t^{(m)}]' \log h_t^{(m)} + \sum_{m=1}^{M}\sum_{t=1}^{T} f(E_Q[S_t^{(m)}]) \tag{C.47}$$

where

$$\begin{aligned} f(E_Q[S_t^{(m)}]) = & \frac{1}{2M} Y_t' C^{-1} Y_t - Y_t' C^{-1} W^{(m)} E_Q[S_t^{(m)}] + \\ & \frac{1}{2} tr\{diag\{E_Q[S_t^{(m)}]\}(W^{(m)})' C^{-1} W^{(m)}\} + \\ & \frac{1}{2}\sum_{n\neq m} tr\{E_Q[S_t^{(m)}(S_t^{(n)})'](W^{(n)})' C^{-1} W^{(m)}\} \end{aligned} \tag{C.48}$$

which makes the derivative:

$$\begin{aligned} \frac{dKL(Q||P)}{d\log h_\tau^{(l)}} = & -\frac{d\log(Z_Q)}{d\log h_\tau^{(l)}} + \frac{d}{d\log h_\tau^{(l)}} \sum_{m=1}^{M}\sum_{t=1}^{T} E_Q[S_t^{(m)}]' \log h_t^{(m)} + \\ & \sum_{m=1}^{M}\sum_{t=1}^{T} \frac{df}{dE_Q[S_t^{(m)}]} \frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}} \end{aligned} \tag{C.49}$$

where

$$\begin{aligned}
\frac{d\log(Z_Q)}{d\log h_\tau^{(l)}} &= \frac{1}{Z_Q}\frac{dZ_Q}{d\log h_\tau^{(l)}} \\
&= \frac{1}{Z_Q}\frac{d}{d\log h_\tau^{(l)}}\sum_{\{S\}}\prod_{t,m}(\ldots(h_t^{(m)})^{S_t^{(m)}}\ldots) \\
&= \frac{1}{Z_Q}\frac{d}{d\log h_\tau^{(l)}}\sum_{\{S\}}\exp\left(\sum_{t,m}(\ldots S_t^{(m)}\log h_t^{(m)}\ldots)\right) \\
&= \frac{1}{Z_Q}\sum_{\{S\}}\exp\left(\sum_{t,m}(\ldots S_t^{(m)}\log h_t^{(m)}\ldots)\right)S_\tau^{(l)} \\
&= \sum_{\{S\}}Q(S)S_\tau^{(l)} \\
&= E_Q[S_\tau^{(l)}]
\end{aligned} \tag{C.50}$$

and

$$\begin{aligned}
\frac{d}{d\log h_\tau^{(l)}}\sum_{m=1}^{M}\sum_{t=1}^{T}E_Q[S_t^{(m)}]'\log h_t^{(m)} &= \sum_{m=1}^{M}\sum_{t=1}^{T}\frac{d}{d\log h_\tau^{(l)}}\left(E_Q[S_t^{(m)}]'\log h_t^{(m)}\right) \\
&= \sum_{m=1}^{M}\sum_{t=1}^{T}\frac{dE_Q[S_t^{(m)}]'}{d\log h_\tau^{(l)}}\log h_t^{(m)} + E_Q[S_t^{(m)}]'\frac{d\log h_t^{(m)}}{d\log h_\tau^{(l)}} \\
&= \sum_{m=1}^{M}\sum_{t=1}^{T}\left(\frac{dE_Q[S_t^{(m)}]'}{d\log h_\tau^{(l)}}\log h_t^{(m)}\right) + E_Q[S_\tau^{(l)}]'
\end{aligned} \tag{C.51}$$

and

$$\begin{aligned}
\frac{df}{dE_Q[S_t^{(m)}]} &= -(W^{(m)})'C^{-1}Y_t + \frac{1}{2}\Delta\{(W^{(m)})'C^{-1}W^{(m)}\} + \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M}(W^{(m)})'C^{-1}W^{(n)}E_Q[S_t^{(n)}])
\end{aligned} \tag{C.52}$$

Substituting these back in gives:

$$\begin{aligned}
\frac{dKL(Q||P)}{d\log h_\tau^{(l)}} &= -E_Q[S_\tau^{(l)}] + \sum_{m=1}^{M}\sum_{t=1}^{T}\left(\frac{dE_Q[S_t^{(m)}]'}{d\log h_\tau^{(l)}}\log h_t^{(m)}\right) + E_Q[S_\tau^{(l)}]' + \\
&\quad \sum_{m=1}^{M}\sum_{t=1}^{T} -(W^{(m)})'C^{-1}Y_t + \frac{1}{2}\Delta\{(W^{(m)})'C^{-1}W^{(m)}\} + \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M}(W^{(m)})'C^{-1}W^{(n)}E_Q[S_t^{(n)}])\frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}} \\
0 &= \sum_{m=1}^{M}\sum_{t=1}^{T}[(\log h_t^{(m)} - (W^{(m)})'C^{-1}Y_t + \frac{1}{2}\Delta\{(W^{(m)})'C^{-1}W^{(m)}\} + \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M}(W^{(m)})'C^{-1}W^{(n)}E_Q[S_t^{(n)}])\frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}}]
\end{aligned} \tag{C.53}$$

We will solve this equation by setting the terms inside the first term of the product (with $\frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}}$ being the second term) equal to zero. This will clearly find a solution. Since the KL divergence is convex (though we will not prove this), any solution to this equation is a

global optimum. This lets us avoid computing $\frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}}$.

$$\begin{aligned}
0 &= \log h_t^{(m)} - (W^{(m)})' C^{-1} Y_t + \frac{1}{2}\Delta\{(W^{(m)})' C^{-1} W^{(m)}\} + \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M} (W^{(m)})' C^{-1} W^{(n)} E_Q[S_t^{(n)}] \\
\log \hat{h_t^{(m)}} &= (W^{(m)})' C^{-1} Y_t - \frac{1}{2}\Delta\{(W^{(m)})' C^{-1} W^{(m)}\} - \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M} (W^{(m)})' C^{-1} W^{(n)} E_Q[S_t^{(n)}] \\
\hat{h_t^{(m)}} &= \exp[(W^{(m)})' C^{-1} Y_t - \frac{1}{2}\Delta\{(W^{(m)})' C^{-1} W^{(m)}\} - \\
&\quad \frac{1}{2}\sum_{n\neq m}^{M} (W^{(m)})' C^{-1} W^{(n)} E_Q[S_t^{(n)}]] \\
\hat{h_t^{(m)}} &= \exp[(W^{(m)})' C^{-1} \tilde{Y_t^{(m)}} - \frac{1}{2}\Delta\{(W^{(m)})' C^{-1} W^{(m)}\}
\end{aligned} \tag{C.54}$$

where again $\tilde{Y}_t^{(m)} = Y_t - \sum_{n\neq m}^{M} (W^{(n)})' \theta_t^{(n)}$, the residual error in Y_t given the predictions from all the state variables except for m.

At the end of this derivation, we end up with a set of fixed point equations for $h_t^{(m)}$ that depend on $E_Q[S_t^{(m)}]$. The estimation of the variational parameters therefore proceeds iteratively, using the forward-backward algorithm for HMMs to calculate the expected values for each m using the current h values, and then updating the h values, iterating until convergence.

Appendix D

Derivation of Sampling Distributions for HPM-DBNs

In Section 4.4, we presented a Gibbs sampling algorithm for doing inference and learning in HPM-DBNs. On each iteration of the Gibbs algorithm, we sample a new value for each parameter in the set Ψ from its full conditional distribution. In this appendix, we provide the derivations of the full conditional distributions each components of Ψ.

Recall that Ψ and each of its components are proportional to the likelihood times the prior:

$$P(\Psi|\{I,S,Y\}) \quad \propto \quad P(\{I,S,Y\}|\Psi)P(\Psi) \tag{D.1}$$

where the likelihood is:

$$\begin{aligned} P(\{I,S,Y\}|\Psi) \quad = \quad & \prod_{t=1}^{T}\prod_{\pi=1}^{\Pi}[P(S_1^{(\pi)}|I_1^{(\pi)},b^{(\pi)})] \times \\ & \prod_{t=2}^{T}\prod_{\pi=1}^{\Pi}[P(S_t^{(\pi)}|S_{t-1}^{(\pi)},\{I_{t-\Omega(\pi)}^{(\pi)}\},\{S_{t-\delta}^{(\pi'\neq\pi)}\},A^{(\pi)})] \times \\ & \prod_{t=1}^{T}P(Y_t|\{S_t\},W,\Sigma) \end{aligned} \tag{D.2}$$

the log-likelihood is:

$$\begin{aligned}\log P(\{I,S,Y\}|\Psi) &= \sum_{t=1}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_1^{(\pi)}|I_1^{(\pi)},b^{(\pi)})] + \\ &\quad \sum_{t=2}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_t^{(\pi)}|S_{t-1}^{(\pi)},\{I_{t-\Omega(\pi)}^{(\pi)}\},\{S_{t-\delta}^{(\pi'\neq\pi)}\},A^{(\pi)})] + \\ &\quad \sum_{t=1}^{T}\log P(Y_t|\{S_t\},W,\Sigma)\end{aligned} \tag{D.3}$$

and the prior over the set Ψ is:

$$\begin{aligned}P(\Psi) &= P(\Sigma)\prod_{\pi=1}^{\Pi}P(b^{(\pi)})P(A^{(\pi)})P(W^{(\pi)}) \\ &= P(\Sigma)\prod_{\pi=1}^{\Pi}P(b_0^{(\pi)})P(b_1^{(\pi)})P(W^{(\pi)})P(A_{\oslash}^{(\pi)})\prod_{o\in\Omega(\pi)}P(A_o^{(\pi)}) \\ \log P(\Psi) &= \log P(\Sigma)+\sum_{\pi=1}^{\Pi}[\log P(b_0^{(\pi)})+\log P(b_1^{(\pi)})+\log P(W^{(\pi)})+ \\ &\quad \log P(A_{\oslash}^{(\pi)})+\sum_{o\in\Omega(\pi)}\log P(A_o^{(\pi)})]\end{aligned} \tag{D.4}$$

So for some normalizing constant c we have:

$$\begin{aligned}
\log P(\Psi|\{I,S,Y\}) &= c + \sum_{t=1}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_1^{(\pi)}|I_1^{(\pi)}, b^{(\pi)})] + \\
&\quad \sum_{t=2}^{T}\sum_{\pi=1}^{\Pi}[\log P(S_t^{(\pi)}|S_{t-1}^{(\pi)}, \{I_{t-\Omega(\pi)}^{(\pi)}\}, \{S_{t-\delta}^{(\pi'\neq\pi)}\}, A^{(\pi)})] + \\
&\quad \sum_{t=1}^{T}\log P(Y_t|\{S_t\}, W, \Sigma) + \\
&\quad \log P(\Sigma) + \sum_{\pi=1}^{\Pi}[\log P(b_0^{(\pi)}) + \log P(b_1^{(\pi)}) + \log P(W^{(\pi)}) + \\
&\quad \log P(A_{\oslash}^{(\pi)}) + \sum_{o\in\Omega(\pi)}\log P(A_o^{(\pi)})]
\end{aligned} \tag{D.5}$$

Below, we choose priors for each component of Ψ and show how each conditional distribution is derived by grouping terms not involving the parameter of interest into the normalizing constant c.

D.1 Sampling b

Let us focus first on the parameters $b^{(\pi)} = \{b_0^{(\pi)}, b_1^{(\pi)}\}\forall\pi$. Since $b_0^{(\pi)}$ and $b_1^{(\pi)}$ are the parameters of multinomial distributions, we will use the conjugate priors, Dirichlet distributions with parameters $\alpha_0^{(\pi)}$ and $\alpha_1^{(\pi)}$. For $b_0^{(\pi)}$, we have:

$$\begin{aligned}
b_0^{(\pi)} &\sim Dir(\alpha_0^{(\pi)}) \\
P(b_0^{(\pi)}) &= \frac{1}{B(\alpha_0^{(\pi)})}\prod_{j=1}^{D^{(\pi)}}(b_{0,j}^{(\pi)})^{\alpha_{0,j}^{(\pi)}-1} \\
\log P(b_0^{(\pi)}) &= -\log(B(\alpha_0^{(\pi)})) + (\alpha_0^{(\pi)}-1)'\log b_0^{(\pi)}
\end{aligned} \tag{D.6}$$

where $\alpha_0^{(\pi)}$ is a $D^{(\pi)}\times 1$ vector and $B(\cdot)$ is the Beta function. The prior is analogous for $\alpha_1^{(\pi)}$. Letting $\{\Psi \setminus b_0^{(\pi)}\}$ denote all parameters in Ψ except $b_0^{(\pi)}$ and grouping all terms not

involving $b_0^{(\pi)}$ into the normalizing constant c, we have:

$$\begin{aligned}
\log P(b_0^{(\pi)}|\{\Psi \setminus b_0^{(\pi)}\}, \{I, S, Y\}) &= c + \log P(S_1^{(\pi)}|I_1^{(\pi)}, b_0^{(\pi)}, b_1^{(\pi)}) + \log P(b_0^{(\pi)}) \\
&= c + (1 - I_1^{(\pi)})(S_1^{(\pi)})' \log b_0^{(\pi)} + I_1^{(\pi)}(S_1^{(\pi)})' \log b_1^{(\pi)} + \\
&\quad - \log(B(\alpha_0^{(\pi)})) + (\alpha_0^{(\pi)} - 1)' \log b_0^{(\pi)} \\
&= c + (1 - I_1^{(\pi)})(S_1^{(\pi)})' \log b_0^{(\pi)} + (\alpha_0^{(\pi)} - 1) \log b_0^{(\pi)} \\
&= c + ((1 - I_1^{(\pi)})S_1^{(\pi)} + \alpha_0^{(\pi)} - 1)' \log b_0^{(\pi)} \\
b_0^{(\pi)}|\{\Psi \setminus b_0^{(\pi)}\}, \{I, S, Y\} &\sim Dir((1 - I_1^{(\pi)})S_1^{(\pi)} + \alpha_0^{(\pi)})
\end{aligned} \tag{D.7}$$

A similar derivation for $b_1^{(\pi)}$ yields:

$$b_1^{(\pi)}|\{\Psi \setminus b_1^{(\pi)}\}, \{I, S, Y\} \sim Dir(I_1^{(\pi)}S_1^{(\pi)} + \alpha_1^{(\pi)}) \tag{D.8}$$

D.2 Sampling A

Next consider the transition matrix parameters. For each π we have a known matrix $A_{\oslash}^{(pi)}$ and unknown matrices $A_o^{(\pi)} \forall o \in \Omega(\pi)$. Each of these matrices contains only one free parameter, which is element $(0, 1)$, the probability of chain π transitioning from 0 to 1 under the conditions represented by the particular $A_o^{(\pi)}$. We will call these free parameters $\rho_o^{(\pi)}$. The probability of transitioning from 0 to 0 in these matrices is then $1 - \rho_o^{(\pi)}$, since no other transitions are legal. Then the values of $\rho_o^{(\pi)}$ are the parameters of Binomial distributions over $A_{o,0,1}^{(\pi)}$ and their conjugate priors are Beta distributions. For $\rho_o^{(\pi)}$ we have:

$$\begin{aligned}
\rho_o^{(\pi)} &\sim Beta(\beta_o^{(\pi)}, \gamma_o^{(\pi)}) \\
P(\rho_o^{(\pi)}) &= \frac{1}{B(\beta_o^{(\pi)}, \gamma_o^{(\pi)})} (\rho_o^{(\pi)})^{\beta_o^{(\pi)} - 1} (1 - \rho_o^{(\pi)})^{\gamma_o^{(\pi)} - 1} \\
\log P(\rho_o^{(\pi)}) &= -\log B(\beta_o^{(\pi)}, \gamma_o^{(\pi)}) + (\beta_o^{(\pi)} - 1) \log \rho_o^{(\pi)} + (\gamma_o^{(\pi)} - 1) \log(1 - \rho_o^{(\pi)})
\end{aligned} \tag{D.9}$$

Here, $\beta_o^{(\pi)}$ and $\gamma_o^{(\pi)}$ are the parameters of the distribution, and $B(\cdot)$ is the Beta function again, not to be confused with the Beta distribution. For a particular $\rho_\omega^{(\pi)}$, we collect terms

of the full conditional to get:

$$
\begin{aligned}
\log P(\rho_\omega^{(\pi)}|\{\Psi \setminus \rho_\omega^{(\pi)}\},\{I,S,Y\}) &= c + \sum_{t=2}^{T}[\log P(S_t^{(\pi)}|S_{t-1}^{(\pi)},\{I_{t-\Omega(\pi)}^{(\pi)}\},S_{t-\delta}^{(\pi')},A^{(\pi)})] + \\
&\quad \log P(\rho_\omega^{(\pi)}) \\
&= c + \sum_{t=2}^{T}[\left(1-\delta(S_{t-\delta}^{(\pi')}>0)\sum_o I_{t-o}^{(\pi)}\right)(S_{t-1}^{(\pi)})'\log A_{\oslash}^{(\pi)} S_t^{(\pi)} + \\
&\quad \delta(S_{t-\delta}^{(\pi')}>0)\sum_o \left(I_{t-o}^{(\pi)}(S_{t-1}^{(\pi)})'\log A_o^{(\pi)} S_t^{(\pi)}\right)] - \\
&\quad \log B(\beta_\omega^{(\pi)},\gamma_\omega^{(\pi)}) + (\beta_\omega^{(\pi)}-1)\log\rho_\omega^{(\pi)} + \\
&\quad (\gamma_\omega^{(\pi)}-1)\log(1-\rho_\omega^{(\pi)}) \\
&= c + \sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0)\left(I_{t-\omega}^{(\pi)}(S_{t-1}^{(\pi)})'\log A_\omega^{(\pi)} S_t^{(\pi)}\right)] + \\
&\quad (\beta_\omega^{(\pi)}-1)\log\rho_\omega^{(\pi)} + (\gamma_\omega^{(\pi)}-1)\log(1-\rho_\omega^{(\pi)}) \\
&= c + \sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)}\left(S_{t-1,0}^{(\pi)}\log\rho_\omega^{(\pi)} S_{t,1}^{(\pi)}\right) + \\
&\quad \delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)}\left(S_{t-1,0}^{(\pi)}\log(1-\rho_\omega^{(\pi)}) S_{t,0}^{(\pi)}\right)] + \\
&\quad (\beta_\omega^{(\pi)}-1)\log\rho_\omega^{(\pi)} + (\gamma_\omega^{(\pi)}-1)\log(1-\rho_\omega^{(\pi)}) \\
&= c + \\
&\quad \left(\sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)} S_{t-1,0}^{(\pi)} S_{t,1}^{(\pi)}] + \beta_\omega^{(\pi)} - 1\right)\times \\
&\quad \log\rho_\omega^{(\pi)} + \\
&\quad \left(\sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)} S_{t-1,0}^{(\pi)} S_{t,0}^{(\pi)}] + \gamma_\omega^{(\pi)} - 1\right)\times \\
&\quad \log(1-\rho_\omega^{(\pi)}) \\
\rho_\omega^{(\pi)}|\{\Psi \setminus \rho_\omega^{(\pi)}\},\{I,S,Y\} &\sim Beta(\sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)} S_{t-1,0}^{(\pi)} S_{t,1}^{(\pi)}] + \beta_\omega^{(\pi)}, \\
&\quad \sum_{t=2}^{T}[\delta(S_{t-\delta}^{(\pi')}>0) I_{t-\omega}^{(\pi)} S_{t-1,0}^{(\pi)} S_{t,0}^{(\pi)}] + \gamma_\omega^{(\pi)}) \qquad \text{(D.10)}
\end{aligned}
$$

D.3 Sampling W

We will treat the W matrices one process and one time point at a time. That is, we will work with $W_d^{(\pi)}$, the length V vector that is contributed to the output variable by process π at the d^{th} time point after it begins. (Note: technically $W_d^{(\pi)}$ is $1 \times V$. To avoid tedious notation in the derivation below, when we write $W_d^{(\pi)}$ we will assume that it has already been transposed into a $V \times 1$ column vector.) We define a multivariate Gaussian prior over $W_d^{(\pi)}$ with parameters $\mu_d^{(\pi)}$ (a $V \times 1$ vector) and $\Sigma_d^{(\pi)}$ (a $V \times V$ matrix):

$$\begin{aligned}
W_d^{(\pi)} &\sim N(\mu_d^{(\pi)}, \Sigma_d^{(\pi)}) \\
P(W_d^{(\pi)}) &= \frac{1}{(2\pi)^{V/2}|\Sigma_d^{(\pi)}|^{1/2}} \exp(-\frac{1}{2}(W_d^{(\pi)} - \mu_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}(W_d^{(\pi)} - \mu_d^{(\pi)})) \\
\log P(W_d^{(\pi)}) &= -\log((2\pi)^{V/2}|\Sigma_d^{(\pi)}|^{1/2}) - \frac{1}{2}(W_d^{(\pi)} - \mu_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}(W_d^{(\pi)} - \mu_d^{(\pi)}) \\
&= c - \frac{1}{2}(W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}W_d^{(\pi)} + (W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)} - \frac{1}{2}(\mu_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}
\end{aligned} \tag{D.11}$$

The full conditional distribution is:

$$\begin{aligned}
\log P(W_d^{(\pi)}|\{\Psi \setminus W_d^{(\pi)}\}, \{I, S, Y\}) &= c + \sum_{t=1}^{T}[\log P(Y_t|\{S_t\}, W, \Sigma)] + \log P(W_d^{(\pi)}) \\
&= c + \sum_{t=1}^{T}[(Y_t)'\Sigma^{-1}\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)} - \\
&\quad \frac{1}{2}(\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)})'\Sigma^{-1}\sum_{\pi}(W^{(\pi)})'S_t^{(\pi)}] - \\
&\quad \frac{1}{2}(W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}W_d^{(\pi)} + (W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}
\end{aligned} \tag{D.12}$$

$$
\begin{aligned}
\log P(W_d^{(\pi)}|\{\Psi \setminus W_d^{(\pi)}\},\{I,S,Y\}) &= c + \sum_{t=1}^{T}[(W_d^{(\pi)} S_{t,d}^{(\pi)})' \Sigma^{-1} Y_t] - \\
&\quad \frac{1}{2}\sum_{t=1}^{T}[(W_d^{(\pi)} S_{t,d}^{(\pi)})' \Sigma^{-1} W_d^{(\pi)} S_{t,d}^{(\pi)}] - \\
&\quad \frac{1}{2}\sum_{t=1}^{T}[(W_d^{(\pi)} S_{t,d}^{(\pi)})' \Sigma^{-1} \sum_{\pi' \neq \pi}(W^{(\pi')})' S_t^{(\pi')}] - \\
&\quad \frac{1}{2}(W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1} W_d^{(\pi)} + (W_d^{(\pi)})'(\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)} \\
&= c - \frac{1}{2}(W_d^{(\pi)})' \left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)} \Sigma^{-1} S_{t,d}^{(\pi)}] + (\Sigma_d^{(\pi)})^{-1}\right) W_d^{(\pi)} + \\
&\quad (W_d^{(\pi)})' \times \\
&\quad \left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)} \Sigma^{-1}(Y_t - \frac{1}{2}\sum_{\pi' \neq \pi}(W^{(\pi')})' S_t^{(\pi')})] + (\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}\right) \\
&= c - \frac{1}{2}(W_d^{(\pi)})' \left(\sum_{t=1}^{T}[(S_{t,d}^{(\pi)})^2] \Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right) W_d^{(\pi)} + \\
&\quad (W_d^{(\pi)})' \left(\sum_{t=1}^{T}[(S_{t,d}^{(\pi)})^2] \Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right) \times \\
&\quad \left(\sum_{t=1}^{T}[(S_{t,d}^{(\pi)})^2] \Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right)^{-1} \times \\
&\quad \left(\Sigma^{-1}\sum_{t=1}^{T}[S_{t,d}^{(\pi)}(Y_t - \frac{1}{2}\sum_{\pi' \neq \pi}(W^{(\pi')})' S_t^{(\pi')})] + (\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}\right) \\
W_d^{(\pi)}|\{\Psi \setminus W_d^{(\pi)}\},\{I,S,Y\} &\sim N(\left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)}] \Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right)^{-1} \times \\
&\quad \left(\Sigma^{-1}\sum_{t=1}^{T}[S_{t,d}^{(\pi)}(Y_t - \frac{1}{2}\sum_{\pi' \neq \pi}(W^{(\pi')})' S_t^{(\pi')})] + (\Sigma_d^{(\pi)})^{-1}\mu_d^{(\pi)}\right), \\
&\quad \left(\sum_{t=1}^{T}[S_{t,d}^{(\pi)}] \Sigma^{-1} + (\Sigma_d^{(\pi)})^{-1}\right)^{-1}) \qquad \text{(D.13)}
\end{aligned}
$$

D.4 Sampling Σ

Recall that Σ is a diagonal covariance matrix; that is, it has the values of σ_v^2 on its diagonal and it is zero off-diagonal. This means we can write $P(Y_t|\{S_t\}, \Psi)$ as:

$$P(Y_t|\{S_t\}, \Psi) = \prod_{v=1}^{V} P(Y_{t,v}|\{S_t\}, \Psi) \tag{D.14}$$

where

$$Y_{t,v}|\{S_t\}, \Psi \sim N(\mu_{t,v}, \sigma_v^2)$$

and

$$\mu_{t,v} = \sum_{\pi} (W_v^{(\pi)})' S(\pi)_t \tag{D.15}$$

Below, we derive the full conditional distribution for each σ_v^2 independently. We define the prior over each σ_v^2 to be an inverse-gamma with parameters η_v and ν_v:

$$\begin{aligned} \sigma_v^2 &\sim InvGamma(\eta_v, \nu_v) \\ P(\sigma_v^2) &= \frac{(\nu_v)^{\eta_v}}{\Gamma(\eta_v)} \left(\frac{1}{\sigma_v^2}\right)^{\eta_v+1} \exp(-\frac{\nu_v}{\sigma_v^2}) \\ \log P(\sigma_v^2) &= \eta_v \log \nu_v - \log \Gamma(\eta_v) - (\eta_v + 1) \log \sigma_v^2 - \frac{\nu_v}{\sigma_v^2} \end{aligned} \tag{D.16}$$

The full conditional distribution is then:

$$\begin{aligned} \log P(\sigma_v^2|\{\Psi \setminus \sigma_v^2\}, \{I, S, Y\}) &= c + \sum_{t=1}^{T} [\log P(Y_{t,v}|\{S_t\}, W, \Sigma)] + \log P(\sigma_v^2) \\ &= c + T \log(\sigma_v) - \frac{1}{\sigma_v^2} \sum_{t=1}^{T} [(Y_{t,v} - \mu_{t,v})^2] - \\ &\quad (\eta_v + 1) \log \sigma_v^2 - \frac{\nu_v}{\sigma_v^2} \\ &= c - (\frac{1}{2}T + \eta_v + 1) \log(\sigma_v^2) - (\nu_v + \frac{1}{2} \sum_{t=1}^{T} [(Y_{t,v} - \mu_{t,v})^2]) \frac{1}{\sigma_v^2} \\ \sigma_v^2|\{\Psi \setminus \sigma_v^2\}, \{I, S, Y\} &\sim InvGamma(\frac{1}{2}T + \eta_v, \nu_v + \frac{1}{2} \sum_{t=1}^{T} [(Y_{t,v} - \mu_{t,v})^2]) \end{aligned} \tag{D.17}$$

Appendix E

Variational Inference for HPM-DBNs

Here we adapt the structured variational approximation for fHMMs (see the tutorial in Section C.2.3 for more details) to the model described in Chapter 4 (with process ordering constraints). Since the exact structure of the model varies with different modeling choices (e.g. the length of influence of an input variable on a process chain) we will work from an example, shown in Figure E.1, which we will approximate as shown in Figure E.2.

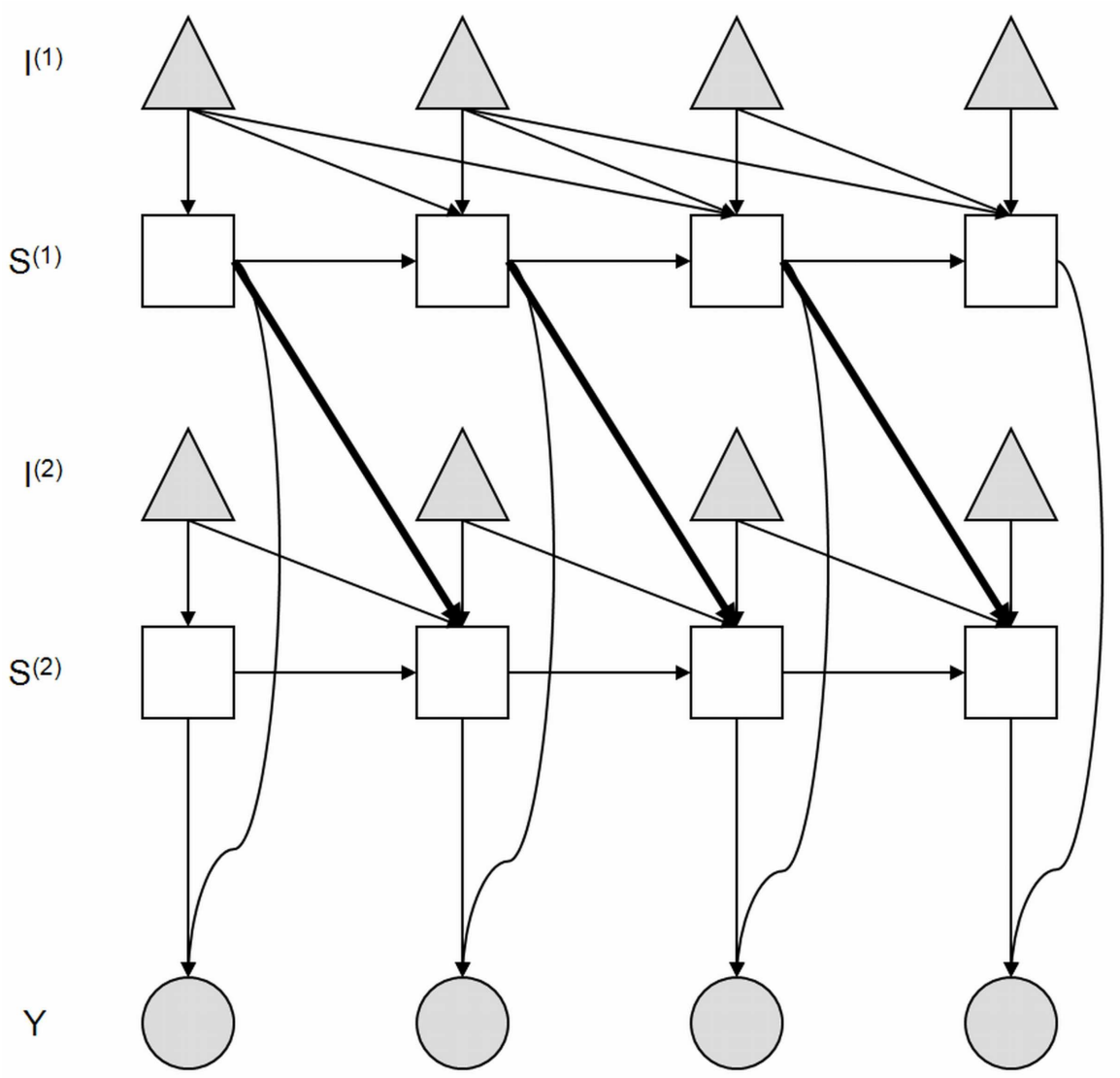

Figure E.1: Example HPM-DBN with process ordering constraints for which the variational inference algorithm is developed. This is the graphical structure of the probability distribution P in the text.

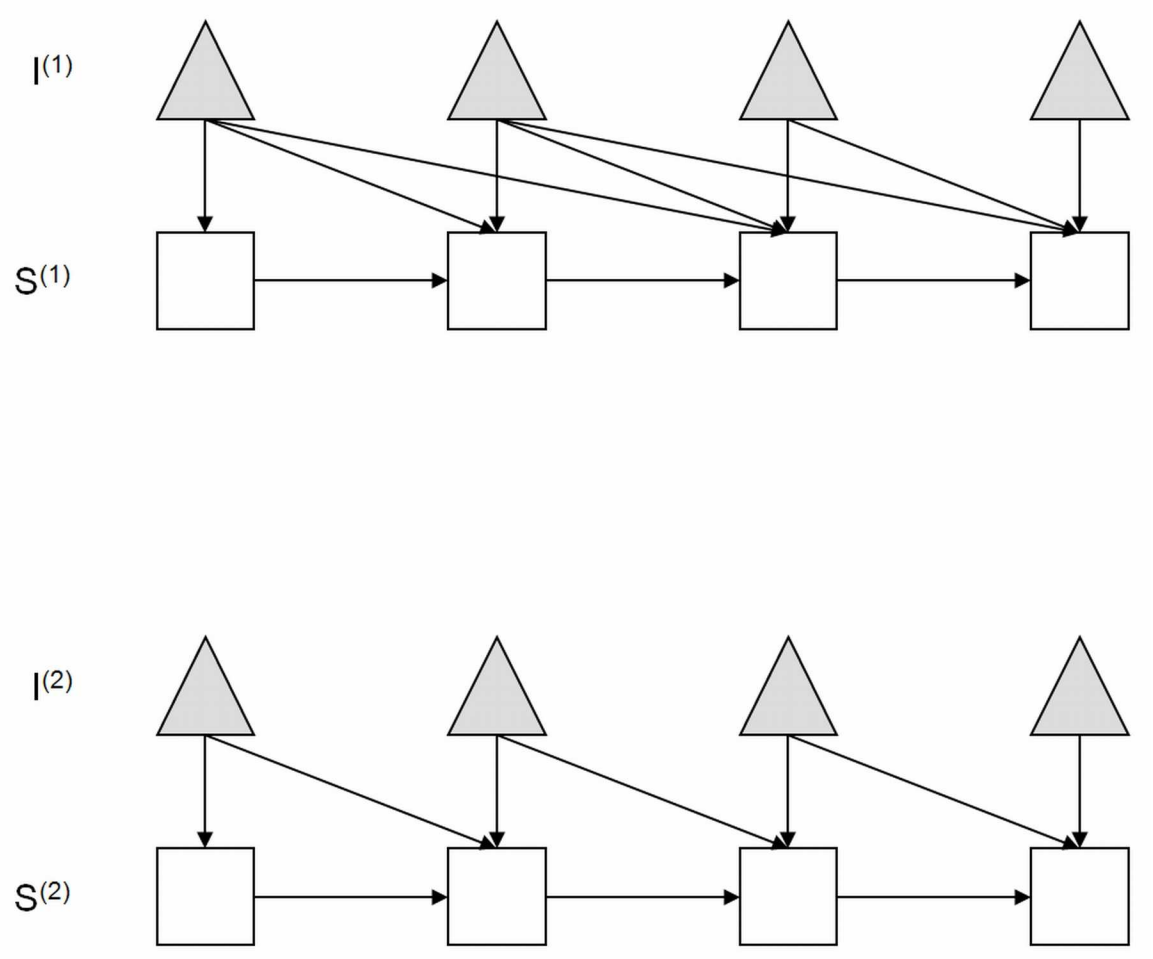

Figure E.2: The structured approximation Q to the true HPM-DBN (P) shown in Figure E.1.

We begin by writing down the probability distribution for the true model (Figure E.1):

$$\begin{aligned} P(\{I,S,Y\}) &= [\prod_{t=1}^{T}\prod_{m=1}^{M} P(I_t^{(m)})] \times \\ & \prod_{m=1}^{M} P(S_1^{(m)}|I_1^{(m)}) \prod_{t=2}^{T} P(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}, \{S_{t-1}^{(n\neq m)}\}) \\ & [\prod_{t=1}^{T} P(Y_t|S_t^{(1)}, S_t^{(2)})] \end{aligned} \tag{E.1}$$

The components of this distribution are as follows (they are explained in more detail in Chapter 4):

$$\begin{aligned} P(I_t^{(m)}) &= (p^{(m)})^{I_t^{(m)}}(1-p^{(m)})^{(1-I_t^{(m)})} \\ \log P(I_t^{(m)}) &= I_t^{(m)} \log p^{(m)} + (1-I_t^{(m)}) \log(1-p^{(m)}) \end{aligned} \tag{E.2}$$

$$\begin{aligned} P(S_1^{(m)}|I_1^{(m)}) &= \left(\prod_k (\pi_{0,k}^{(m)})^{S_{1,k}^{(m)}}\right)^{(1-I_1^{(m)})} \left(\prod_k (\pi_{1,k}^{(m)})^{S_{1,k}^{(m)}}\right)^{I_1^{(m)}} \\ \log P(S_1^{(m)}|I_1^{(m)}) &= (1-I_1^{(m)})(S_1^{(m)})' \log \pi_0^{(m)} + I_1^{(m)}(S_1^{(m)})' \log \pi_1^{(m)} \end{aligned} \tag{E.3}$$

$$
\begin{aligned}
P(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}, \{S_{t-1}^{(n\neq m)}\}) &= \left(\prod_{i,j}(A_{\oslash,i,j}^{(m)})^{S_{t-1,i}^{(m)}S_{t,j}^{(m)}}\right)^{1-\delta(S_{t-1}^{(n)}>0)\sum_o I_{t-o}^{(m)}} \times \\
&\quad \prod_o\left(\left(\prod_{i,j}(A_{o,i,j}^{(m)})^{S_{t-1,i}^{(m)}S_{t,j}^{(m)}}\right)^{\delta(S_{t-1}^{(n)}>0)I_{t-o}^{(m)}}\right) \\
\log P(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}, \{S_{t-1}^{(n\neq m)}\}) &= \left(1-\delta(S_{t-1}^{(n)}>0)\sum_o I_{t-o}^{(m)}\right) \times \\
&\quad (S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)} + \\
&\quad \delta(S_{t-1}^{(n)}>0)\sum_o\left(I_{t-o}^{(m)}(S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)}\right)
\end{aligned}
\tag{E.4}
$$

The form of $P(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}, \{S_\tau^{(n\neq m)}\})$ varies depending on specific modeling choices. Generally, $S_t^{(m)}$ can depend on some number of input values $I_{t-\Omega(m)}^{(m)}$ where $\Omega(m)$ is a vector of the legal offset values for process m. The length and values of these vectors vary from one process to another, and therefore from one chain to another. $S_t^{(m)}$ can also depend on zero or more other processes ($n \neq m$). These dependencies encode process ordering constraints. In this example, chain 2 at time t depends on chain 1 at time $t-1$ but in general the dependence could be on any time point of any other chain. $\{S_{t-1}^{(n\neq m)}\}$ can also be the empty set, as is the case for chain 1, which does not depend on any other chains. Here, $A_{\oslash}^{(m)}$ contains the transition dynamics for the case where the relevant input values are 0 and/or a chain on which m depends has not yet begun. $A_o^{(m)}$ encodes the dynamics for the case where the input value at $t-o$ was 1 and any dependencies on other chains are satisfied.

$$
\begin{aligned}
P(Y_t|S_t) &= |C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}} \times \\
&\quad \exp\left(-\frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1} (Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})\right) \\
\log P(Y_t|S_t) &= \log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) - \\
&\quad \frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1} (Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})
\end{aligned}
\tag{E.5}
$$

Putting all this together gives:

$$
\begin{aligned}
\log P(\{I, S, Y\}) &= \sum_{m=1}^{M}\sum_{t=1}^{T} \log P(I_t^{(m)}) + \sum_{m=1}^{M} \log P(S_1^{(m)}|I_1^{(m)}) + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T} \log P(S_t^{(m)}|S_{t-1}^{(m)}, \{I^{(m)}\}, \{S^{(n)}\}) + \sum_{t=1}^{T} \log P(Y_t|S_t^{(1)}, S_t^{(2)}) \\
&= \sum_{m=1}^{M}\sum_{t=1}^{T} I_t^{(m)} \log p^{(m)} + (1 - I_t^{(m)}) \log(1 - p^{(m)}) + \\
&\quad \sum_{m=1}^{M} (1 - I_1^{(m)})(S_1^{(m)})' \log \pi_{\oslash}^{(m)} + I_1^{(m)} (S_1^{(m)})' \log \pi^{(m)} + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T} [\left(1\delta(S_{t-1}^{(n)} > 0) \sum_{o} I_{t-o}^{(m)}\right) \times \\
&\quad (S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)} + \\
&\quad \delta(S_{t-1}^{(n)} > 0) \sum_{o} \left(I_{t-o}^{(m)} (S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)}\right)] + \\
&\quad \sum_{t=1}^{T} \log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) - \frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1} (Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})
\end{aligned}
\tag{E.6}
$$

The approximating distribution (Figure E.2) is:

$$\begin{aligned} Q(\{I,S\}) &= [\prod_{t=1}^{T}\prod_{m=1}^{M} P(I_t^{(m)})] \times \\ & \quad [\prod_{m=1}^{M} Q(S_1^{(m)}|I_1^{(m)}) \prod_{t=2}^{T} Q(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\})] \end{aligned} \tag{E.7}$$

The components of Q are similar to those of P except for the addition of variational parameters h to replace the dependence on outputs Y and the process ordering constraints:

$$\begin{aligned} Q(S_1^{(m)}|I_1^{(m)}) &= \left(\prod_k (h_{1,k}^{(m)}\pi_{0,k}^{(m)})^{S_{1,k}^{(m)}}\right)^{(1-I_1^{(m)})} \left(\prod_k (h_{1,k}^{(m)}\pi_{1,k}^{(m)})^{S_{1,k}^{(m)}}\right)^{I_1^{(m)}} \\ \log Q(S_1^{(m)}|I_1^{(m)}) &= (S_1^{(m)})' \log h_1^{(m)} + (1-I_1^{(m)})(S_1^{(m)})' \log \pi_0^{(m)} + I_1^{(m)}(S_1^{(m)})' \log \pi_1^{(m)} \end{aligned} \tag{E.8}$$

$$\begin{aligned} Q(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}) &= \left(\prod_j (h_{t,j}^{(m)} \prod_i (A_{\oslash,i,j}^{(m)})^{S_{t-1,i}^{(m)}})^{S_{t,j}^{(m)}}\right)^{1-\sum_o I_{t-o}^{(m)}} \times \\ & \quad \prod_o \left(\left(\prod_j (h_{t,j}^{(m)} \prod_i (A_{o,i,j}^{(m)})^{S_{t-1,i}^{(m)}})^{S_{t,j}^{(m)}}\right)^{I_{t-o}^{(m)}}\right) \\ \log Q(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\}) &= (S_t^{(m)})' \log h_t^{(m)} + \left(1 - \sum_o I_{t-o}^{(m)}\right) (S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)} + \\ & \quad \sum_o I_{t-o}^{(m)} (S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)} \end{aligned} \tag{E.9}$$

Note that in the structured approximation, there is no dependence on other chains in $Q(S_t^{(m)}|S_{t-1}^{(m)}, \{I_{t-\Omega(m)}^{(m)}\})$. The full distribution is then:

$$
\begin{aligned}
\log Q(\{I,S\}) &= \sum_{t=1}^{T}\sum_{m=1}^{M}\log P(I_t^{(m)}) + \\
&\quad \sum_{m=1}^{M}\log Q(S_1^{(m)}|I_1^{(m)}) + \sum_{m=1}^{M}\sum_{t=2}^{T}\log Q(S_t^{(t)}|S_{t-1}^{(m)},\{I^{(m)}) \\
&= \sum_{t=1}^{T}\sum_{m=1}^{M} I_t^{(m)}\log p^{(m)} + (1-I_t^{(m)})\log(1-p^{(m)}) + \\
&\quad \sum_{m=1}^{M}(S_1^{(m)})'\log h_1^{(m)} + (1-I_1^{(m)})(S_1^{(m)})'\log\pi_{\oslash}^{(m)} + I_1^{(m)}(S_1^{(m)})'\log\pi^{(m)} \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T}(S_t^{(m)})'\log h_t^{(m)} + \left(1-\sum_o I_{t-o}^{(m)}\right)(S_{t-1}^{(m)})'\log A_{\oslash}^{(m)}S_t^{(m)} + \\
&\quad \sum_o I_{t-o}^{(m)}(S_{t-1}^{(m)})'\log A_o^{(m)}S_t^{(m)}
\end{aligned}
\tag{E.10}
$$

As discussed in the overview of variational inference (Section C.2.1), we wish to choose paramters of Q to minimize the KL divergence between P and Q. The KL divergence is:

$$
\begin{aligned}
KL(Q||P) &= -\log Z_Q + \log Z_P + \sum_{\{S\}} Q(\{I,S\})[\log Q(\{I,S\}) - \log P(\{I,S,Y\})] \\
&= -\log Z_Q + \log Z_P + \sum_{\{S\}} Q(\{I,S\})[\\
&\quad [\sum_{t=1}^{T}\sum_{m=1}^{M}(I_t^{(m)} \log p^{(m)} + (1 - I_t^{(m)}) \log(1 - p^{(m)})) + \\
&\quad \sum_{m=1}^{M}((S_1^{(m)})' \log h_1^{(m)} + (1 - I_1^{(m)})(S_1^{(m)})' \log \pi_0^{(m)} + I_1^{(m)}(S_1^{(m)})' \log \pi_1^{(m)}) + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T}((S_t^{(m)})' \log h_t^{(m)} + \left(1 - \sum_o I_{t-o}^{(m)}\right)(S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)} + \\
&\quad \sum_o I_{t-o}^{(m)}(S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)})] - \\
&\quad [\sum_{m=1}^{M}\sum_{t=1}^{T}(I_t^{(m)} \log p^{(m)} + (1 - I_t^{(m)}) \log(1 - p^{(m)})) + \\
&\quad \sum_{m=1}^{M}((1 - I_1^{(m)})(S_1^{(m)})' \log \pi_0^{(m)} + I_1^{(m)}(S_1^{(m)})' \log \pi_1^{(m)}) + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T}(\left(1 - \delta(S_{t-1}^{(n)} > 0) \sum_o I_{t-o}^{(m)}\right) \times \\
&\quad (S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)} + \\
&\quad \delta(S_{t-1}^{(n)} > 0) \sum_o \left(I_{t-o}^{(m)}(S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)}\right)) + \\
&\quad \sum_{t=1}^{T}(\log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) - \frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)}))]]
\end{aligned}
\tag{E.11}
$$

After simplifying, this is:

$$
\begin{aligned}
KL(Q||P) &= -\log Z_Q + \log Z_P + \sum_{\{S\}} Q(\{I,S\})[\\
&\quad \sum_{m=1}^{M}\sum_{t=1}^{T}((S_t^{(m)})' \log h_t^{(m)} + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T}\delta(S_{t-1}^{(n)}=0)(\sum_o I_{t-o}^{(m)}(S_{t-1}^{(m)})' \log A_o^{(m)} S_t^{(m)} - \\
&\quad (\sum_o I_{t-o}^{(m)})(S_{t-1}^{(m)})' \log A_{\oslash}^{(m)} S_t^{(m)}) - \\
&\quad T\log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) + \\
&\quad \sum_{t=1}^{T}\frac{1}{2}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})] \\
&= -\log Z_Q + \log Z_P + \sum_{m=1}^{M}\sum_{t=1}^{T}((E_Q[S_t^{(m)}])' \log h_t^{(m)} + \\
&\quad \sum_{m=1}^{M}\sum_{t=2}^{T}Q(S_{t-1}^{(n)}=0)(\sum_o I_{t-o}^{(m)} tr(\log A_o^{(m)} E_Q[S_t^{(m)}(S_{t-1}^{(m)})']) - \\
&\quad (\sum_o I_{t-o}^{(m)}) tr(\log A_{\oslash}^{(m)} E_Q[S_t^{(m)}(S_{t-1}^{(m)})'])) - \\
&\quad T\log(|C|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}) + \\
&\quad \sum_{t=1}^{T}\frac{1}{2}E_Q[(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})' C^{-1}(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)})]
\end{aligned}
\tag{E.12}
$$

We wish to set the derivative of the KL divergence with respect to the variational parameters equal to zero and solve for their updates.

$$\begin{aligned}\frac{dKL(Q||P)}{d\log h_\tau^{(l)}} = 0 \quad = \quad & -E_Q[S_\tau^{(l)}] + \sum_{m=1}^{M}\sum_{t=1}^{T}\left(\frac{dE_Q[S_t^{(m)}]'}{d\log h_\tau^{(l)}}\log h_t^{(m)}\right) + E_Q[S_\tau^{(l)}]' + \\ & \sum_{m=1}^{M}\sum_{t=1}^{T} -(W^{(m)})'C^{-1}Y_t + \frac{1}{2}\Delta\{(W^{(m)})'C^{-1}W^{(m)}\} + \\ & \frac{1}{2}\sum_{n\neq m}^{M}(W^{(m)})'C^{-1}W^{(n)}E_Q[S_t^{(n)}])\frac{dE_Q[S_t^{(m)}]}{d\log h_\tau^{(l)}} + \\ & \frac{d}{d\log h_\tau^{(l)}}[\sum_{m=1}^{M}\sum_{t=2}^{T}Q(S_{t-1}^{(n)}=0)(\sum_{o}I_{t-o}^{(m)}tr(\log A_o^{(m)}E_Q[S_t^{(m)}(S_{t-1}^{(m)})'])- \\ & (\sum_{o}I_{t-o}^{(m)})tr(\log A_\oslash^{(m)}E_Q[S_t^{(m)}(S_{t-1}^{(m)})']))]\end{aligned} \tag{E.13}$$

We can solve this equation for the parameters h using an iterative method.

Appendix F

Visualizations of Learned HPM Processes

This appendix contains images of the processes of HPM-3-K learned using baseline, regularized, and basis function versions of HPMs on Participant D. Each section corresponds to one of the methods for parameterizing and training HPMs, and the three figures in each section show the ViewPicture, ReadSentence, and Decide process response signatures averaged over time. Darker red voxels, therefore, have higher amplitude on average. Details on HPM-3-K are found in Section 5.2.1.

F.1 Standard HPMs

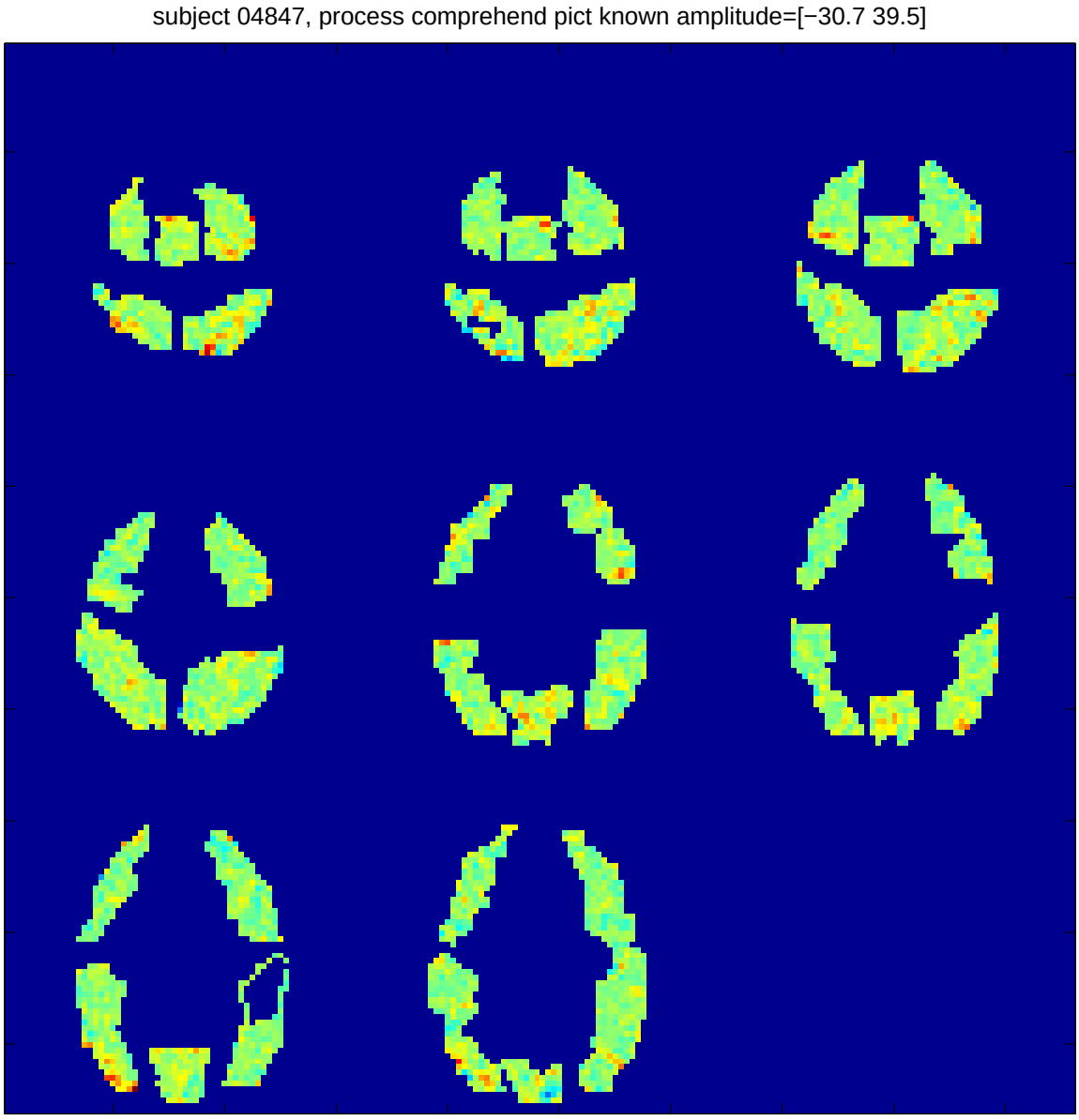

Figure F.1: Learned ViewPicture process in HPM-3-K learned from all trials and all voxels using standard HPMs.

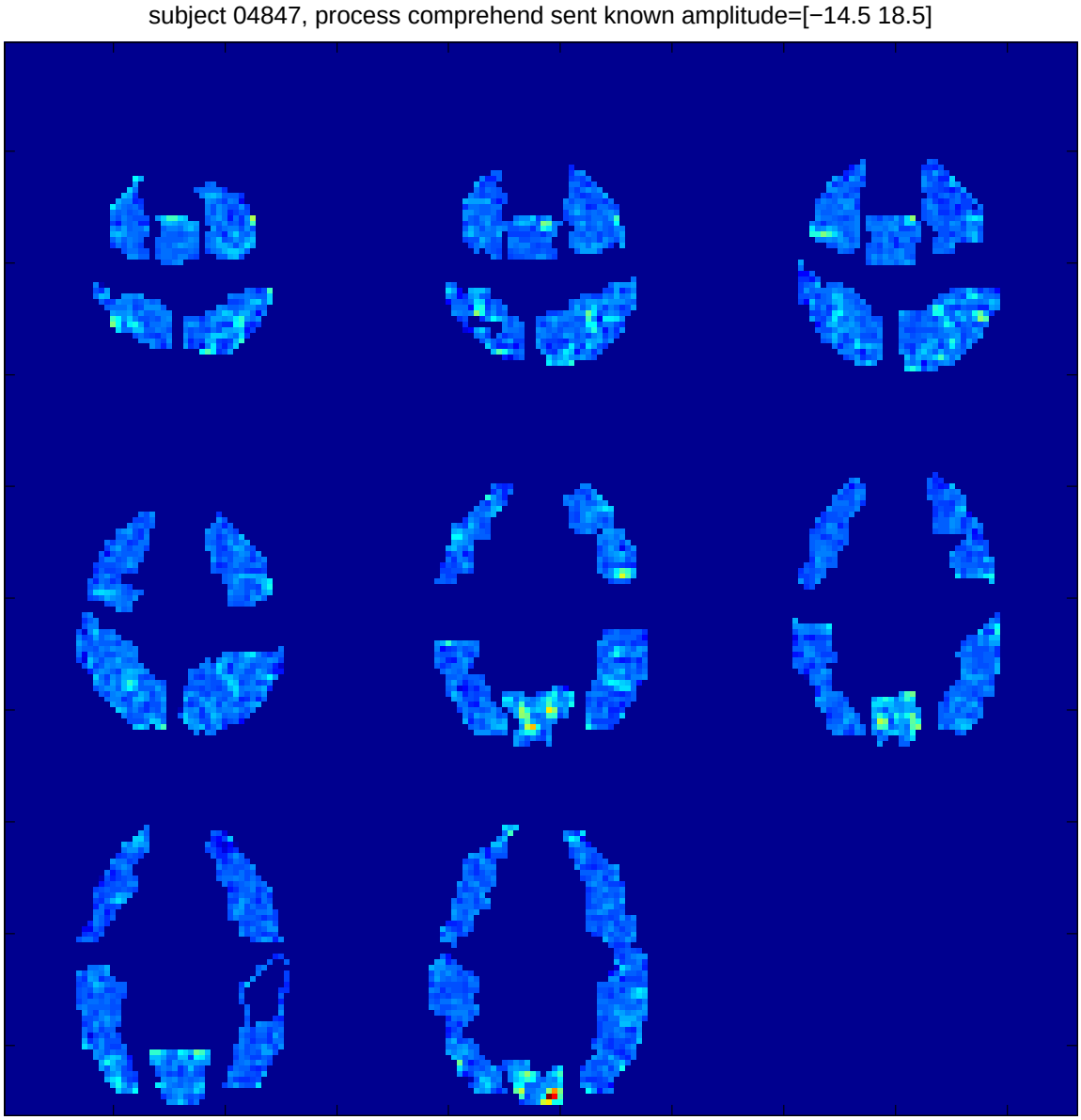

Figure F.2: Learned ReadSentence process in HPM-3-K learned from all trials and all voxels using standard HPMs.

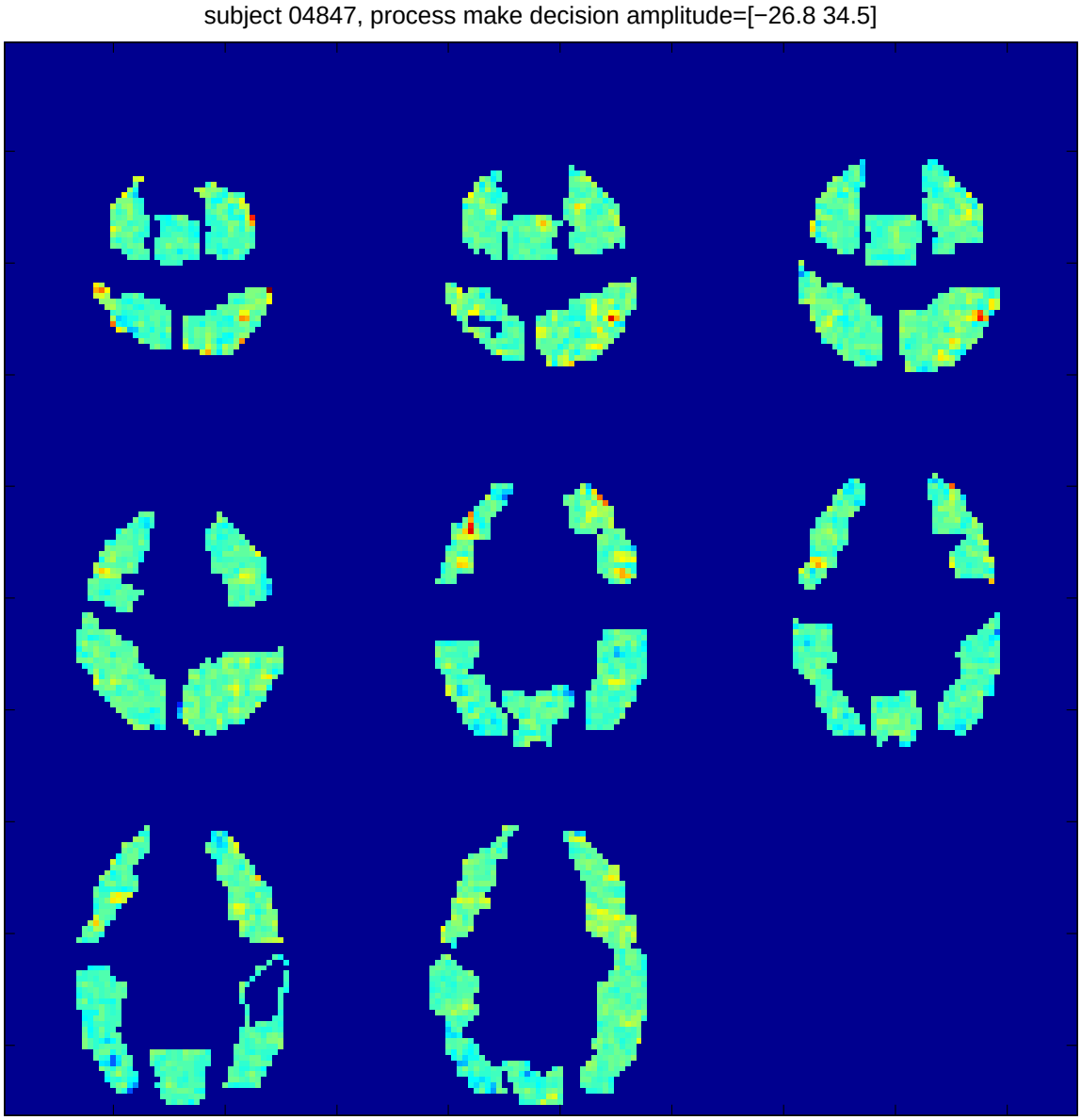

Figure F.3: Learned Decide process in HPM-3-K learned from all trials and all voxels using standard HPMs.

F.2 Regularized HPMs

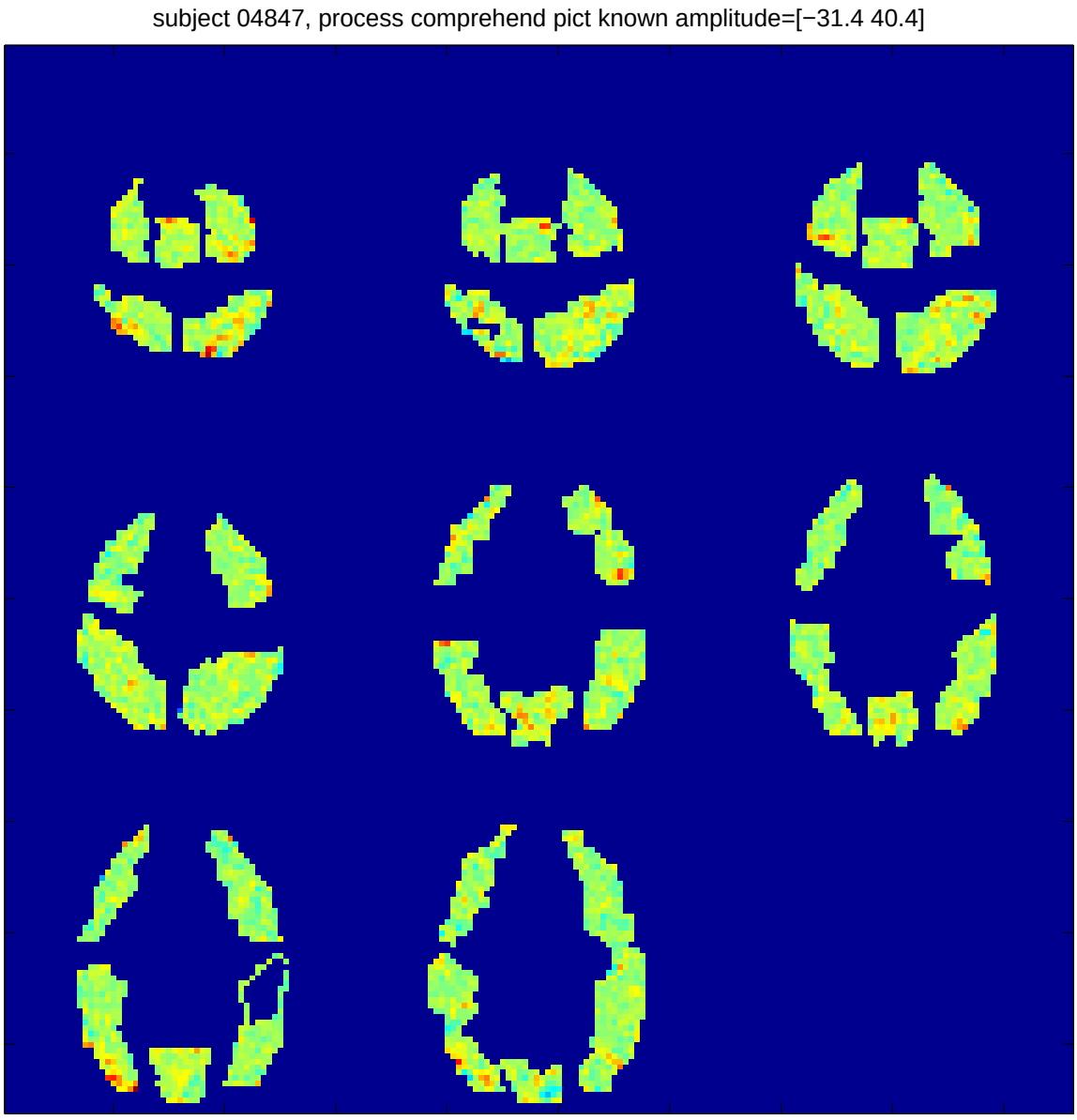

Figure F.4: Learned ViewPicture process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs.

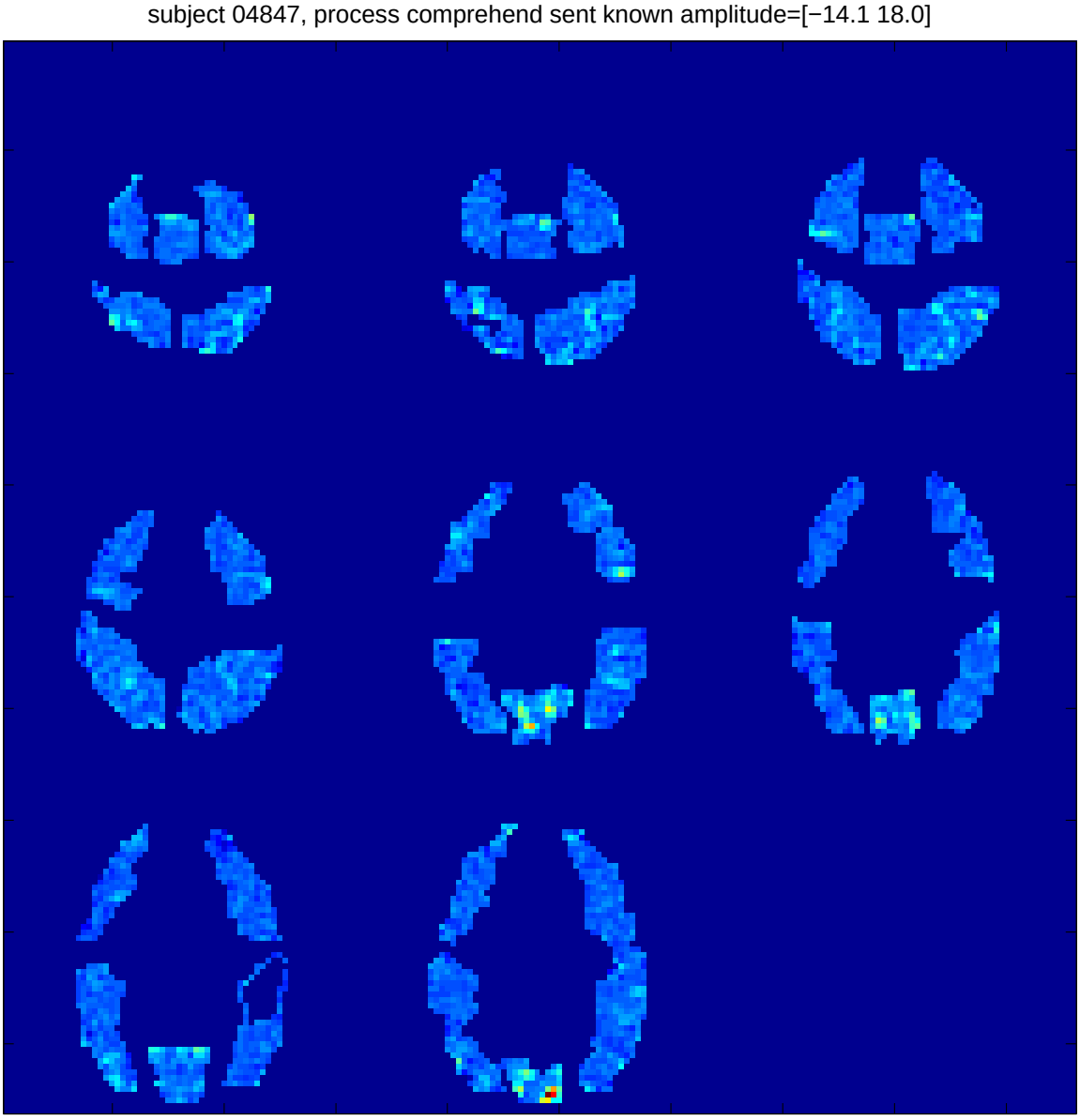

Figure F.5: Learned ReadSentence process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs.

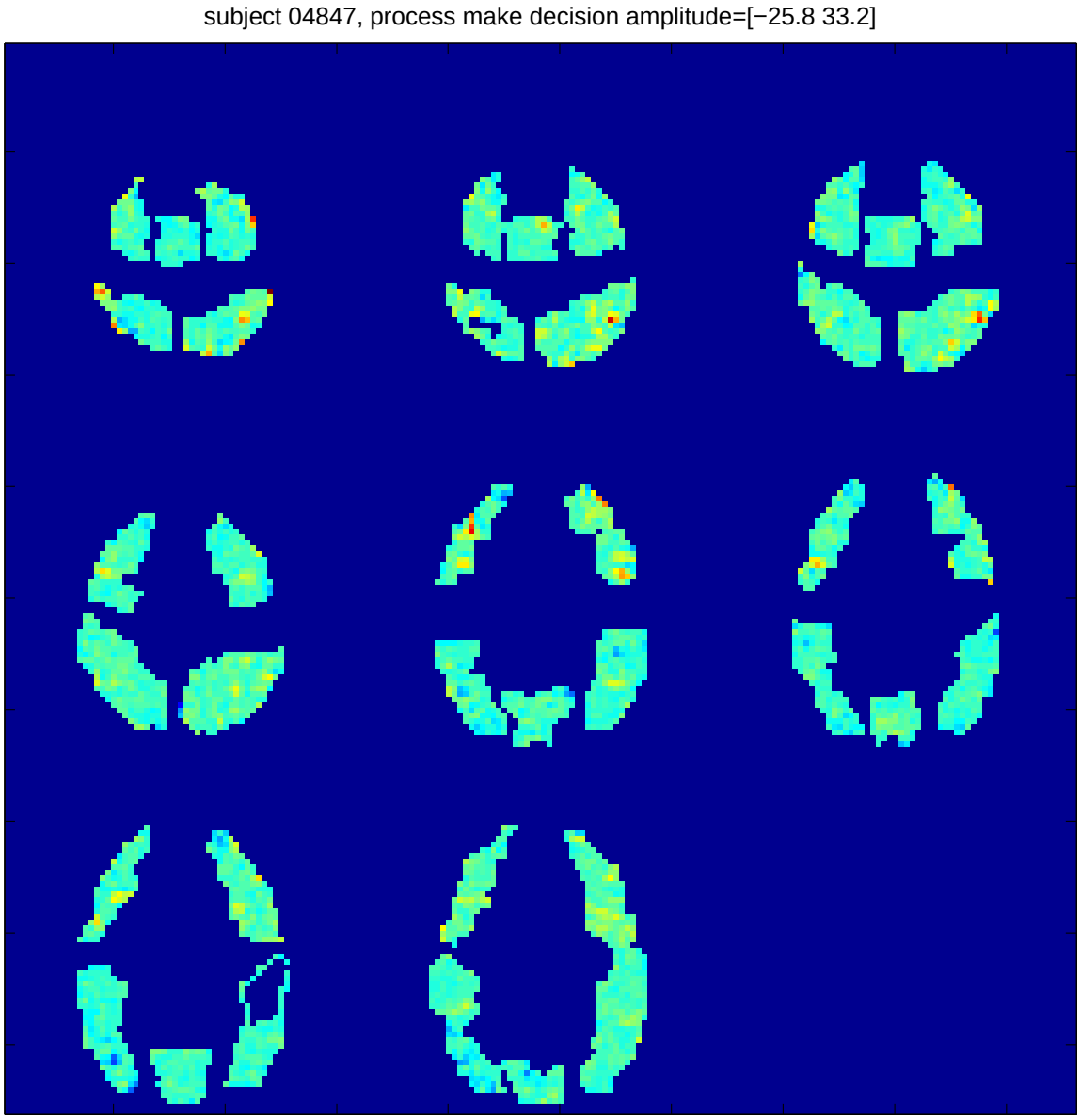

Figure F.6: Learned Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using regularized (for spatial and temporal smoothness) HPMs.

F.3 HPMs with Basis Functions

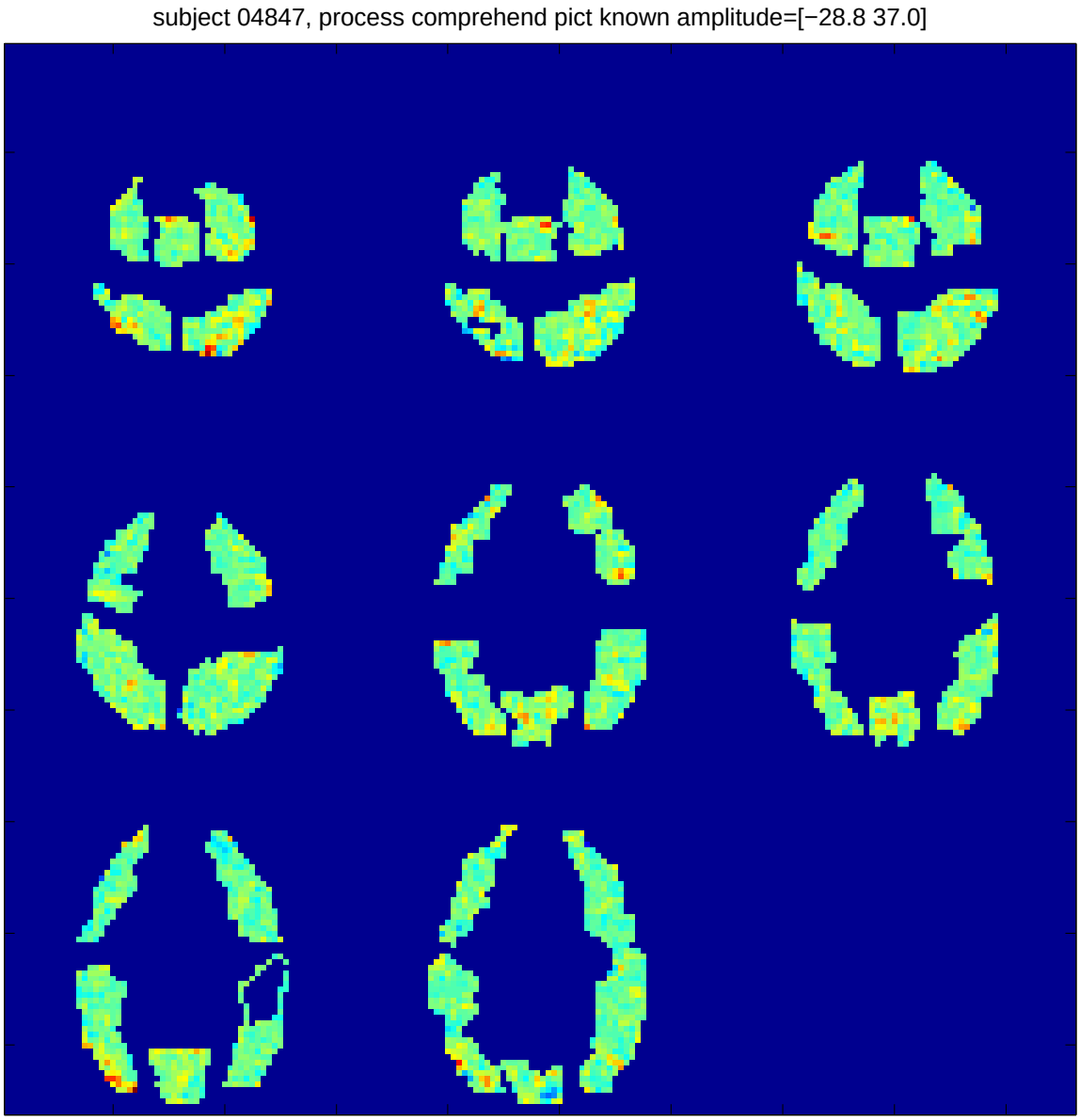

Figure F.7: Learned ViewPicture process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions.

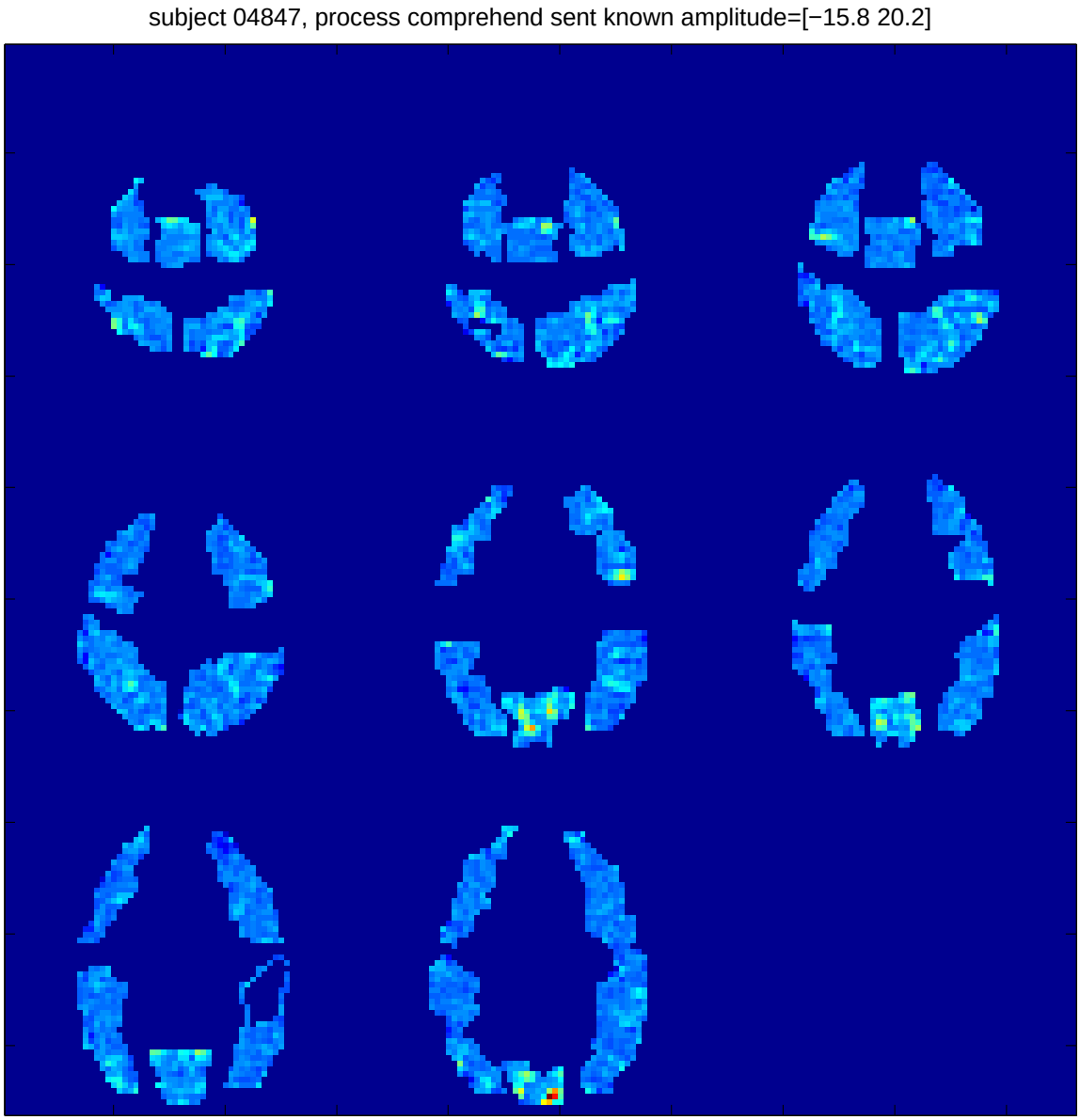

Figure F.8: Learned ReadSentence process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions.

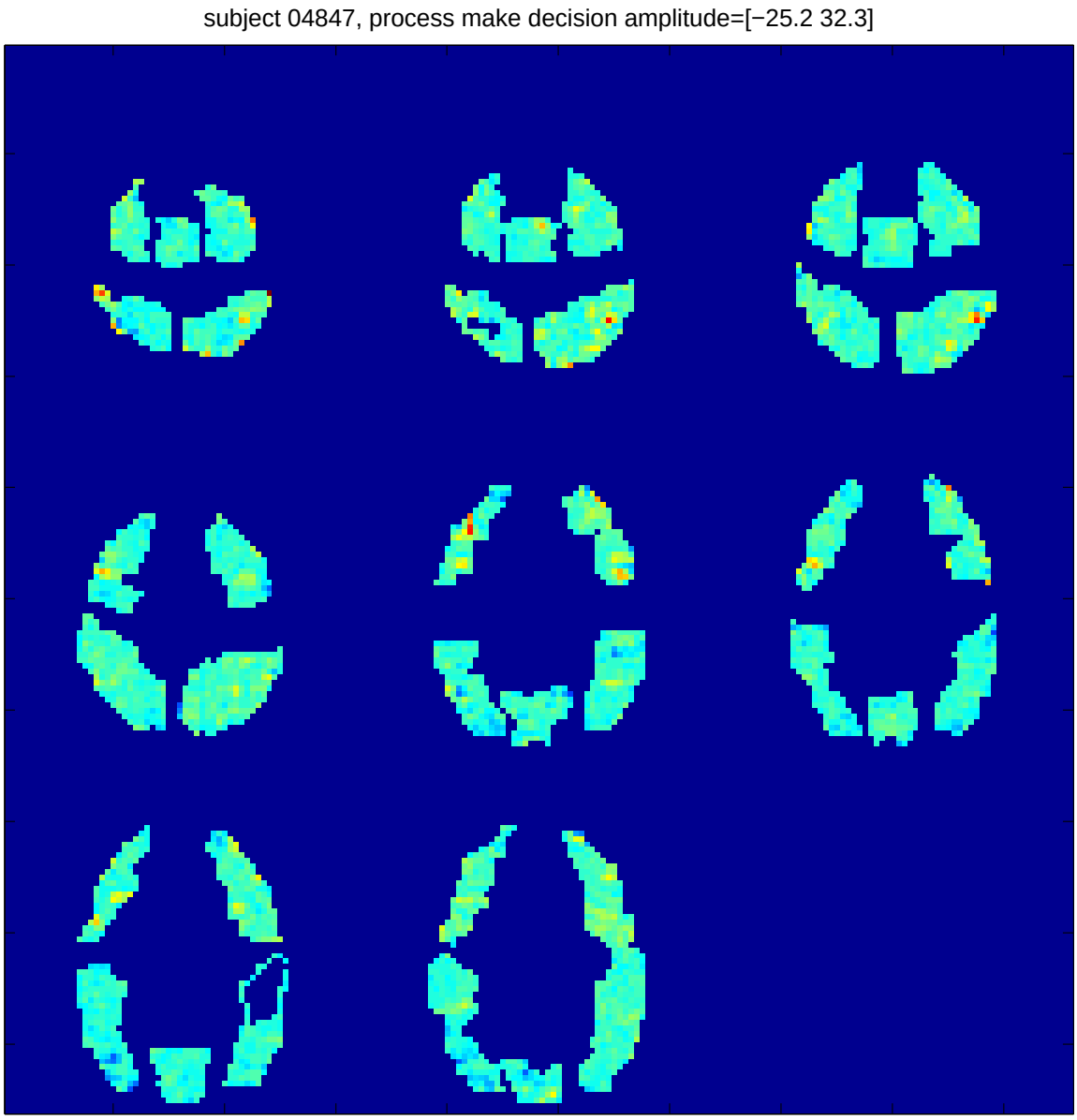

Figure F.9: Learned Decide process in HPM-3-K learned from all trials and all voxels of Particpant D using HPMs with basis functions.

Bibliography

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 2008. 3.1.3

Rasmus M. Birn, Ziad S. Saad, and Peter A. Bandettini. Spatial heterogeneity of the nonlinear dynamics in the FMRI BOLD response. *NeuroImage*, 14:817–826, 2001. 1.3

Geoffrey M. Boynton, Stephen A. Engel, Gary H. Glover, and David J. Heeger. Linear systems analysis of functional magnetic resonance imaging in human V1. *The Journal of Neuroscience*, 16(13):4207–4221, 1996. 1.3, 2.1, 3.3, 5.1.1

Philippe Ciuciu, Jean-Baptiste Poline, Guillaume Marrelc, Jerome Idier, Christophe Pallier, and Habib Benali. Unsupervised robust nonparametric estimation of the hemodynamic response function for any fmri experiment. *IEEE Transactions on Medical Imaging*, 22(10):1235–1251, October 2003. 1.3

David D. Cox and Robert L. Savoy. Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19:261–270, 2003. 1.3

Anders M. Dale. Optimal experimental design for event-related fMRI. *Human Brain Mapping*, 8:109–114, 1999. 1.3, 2.2.2, 6.3

Anders M. Dale and Randy L. Buckner. Selective averaging of rapidly presented individual trials using fMRI. *Human Brain Mapping*, 5:329–340, 1997. 1.3, 6.3

Thomas Dean and Keiji Kanazawa. Probabilistic temporal reasoning. In *Proceedings AAAI-88*, pages 524–528, Cambridge, Massachusetts, 1988. AAAI, MIT Press. 1.3

Thomas Dean and Michael Wellman. *Planning and Control*. Morgan Kaufmann Publishers, San Francisco, California, 1991. 1.3

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, 1977. 2.2.3

William F. Eddy, Mark Fitzgerald, Christopher Genovese, Nicole Lazar, Audris Mockus, and Joel Welling. The challenge of functional magnetic resonance imaging. *Journal of Computational and Graphical Statistics*, 8(3):545–558, 1999. 5.2

S. Faisan, L. Thoraval, J.-P. Armspach, and F. Heitz. Hidden markov multiple event sequence models: A paradigm for the spatio-temporal analysis of fmri data. *Medical Image Analysis*, 11:1–20, 2007. 1.3

Elia Formisano and Rainer Goebel. Tracking cognitive processes with functional mri mental chronometry. *Current Opinion in Neurobiology*, 13:174–181, 2003. 1.3

Karl J. Friston. *Human brain function*, chapter Introduction to Statistical Parametric Mapping (K. Friston). Academic Press, 2nd edition, 2003. *http://www.fil.ion.ucl.ac.uk/spm/doc/intro/*. 1.3, 6.3

Karl J. Friston, Andrea Mechelli, Robert Turner, and Cathy J. Price. Nonlinear responses in fmri: The balloon model, volterra kernels, and other hemodynamics. *NeuroImage*, 12:466–477, 2000. 1.3

Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997. (document), 1.3, 4.1, 4.1.1, C, C.1, C.2.1, C.2.2, C.2, C.2.3, C.3

James V. Haxby, M. Ida Gobbini, Maura L. Furey, Alumit Ishai, Jennifer L. Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293:2425–2430, September 2001. 1.3

John-Dylan Haynes and Geraint Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7:523–534, July 2006. 1.3

Rik N.A. Henson, Cathy J. Price, Michael D. Rugg, Robert Turner, and Karl J. Friston. Detecting latency differences in event-related bold responses: Application to words versus nonwords, and initial versus repeated face presentations. *Neuroimage*, 15:83–97, 2002. 1.3

Gholam-Ali Hossein-Zadeh, Babak A. Ardekani, and Hamid Soltanian-Zadeh. A signal subspace approach for modeling the hemodynamic response function in fmri. *Magnetic Resonance Imaging*, 21:835–843, 2003. 1.3, 3.2, 5.2.2

Rebecca A. Hutchinson, Tom M. Mitchell, and Indrayana Rustandi. Hidden process models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 433–440, 2006. 1.3

Rebecca A. Hutchinson, Tom M. Mitchell, Radu Stefan Niculescu, and Indrayana Rustandi. Modeling fmri data generated by overlapping cognitive processes with unknown onsets using hidden process models. *Neuroimage*, 46(1):87–104, 2009. 1.3

Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002. 1.3

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 105–161. Kluwer Academic Publishers, 1998. C.2.1

Yukiyasu Kamitani and Frank Tong. Decoding the visual and subjective contents of the human brain. *Nature Neuroscience*, 8:679–685, 2005. 1.3

Kendrick N. Kay, Thomas Naselaris, Ryan J. Prenger, and Jack L. Gallant. Identifying natural images from human brain activity. *Nature*, 452:352–355, 2008. 1.3

Tim A. Keller, Marcel Adam Just, and V. Andrew Stenger. Reading span and the time-course of cortical activation in sentence-picture verification. In *Annual Convention of the Psychonomic Society*, 2001. 1.2, 5.2

Chien Heng Liao, Keith J. Worsley, Jean-Baptiste Poline, John A.D. Aston, Gary H. Duncan, and Alan C. Evans. Estimating the delay of the fmri response. *Neuroimage*, 16(3): 593–606, 2002. 1.3

Martin A. Lindquist, Christian Waugh, and Tor D. Wager. Modeling state-related fmri activity using change-point theory. *Neuroimage*, 35:1125–1141, 2007. 1.3

David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 3.3, 4.3, 4.4

David J.C. MacKay. Introduction to Monte Carlo methods. In Michael I. Jordan, editor, *Learning In Graphical Models*, pages 175–204. Kluwer Academic Publishers, 1998. 3.3, 4.3, 4.4

Salima Makni, Jerome Idier, Thomas Vincent, Bertrand Thirion, Ghislaine Dehaene-Lambertz, and Philippe Ciuciu. A fully bayesian approach to the parcel-based detection-estimation of brain activity in fmri. *Neuroimage*, 41(3):941–969, 2008. 1.3

Ravi S. Menon, David C. Luknowsky, and Joseph S. Gati. Mental chronometry using latency-resolved functional mri. *Proceedings of the National Academy of Sciences*, 95 (18):10902–10907, 1998. 1.3

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. 2.2.2, 5.2.1

Tom M. Mitchell, Rebecca Hutchinson, Radu Stefan Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004. 1.3

Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L. Malave, Robert A. Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *Science*, 320, 2008. 1.3

Kevin P. Murphy. Dynamic bayesian networks. To appear in *Probabilistic Graphical Models*, M. Jordan, November 2002. 1.3, 4, 4.1

Radu Stefan Niculescu. *Exploiting Parameter Domain Knowledge for Learning in Bayesian Networks*. PhD thesis, Carnegie Mellon University, July 2005. CMU-CS-05-147. 1.3

Radu Stefan Niculescu, Tom M. Mitchell, and R. Bharat Rao. Bayesian network learning with parameter constraints. *JMLR Special Topic on Machine Learning and Large Scale Optimization*, 7:1357–1383, July 2006. 1.3

Mark M. Palatucci and Tom M. Mitchell. Classification in very high dimensional problems with handfuls of examples. In *Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*. Springer-Verlag, September 2007. 1.3

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 1.3, 4.4.1, C, C.1.1

Steven L. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351, March 2002. 4.4, 4.4.1

Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer-Verlag New York, Inc., 2004. C.2.1, C.2.1

Lei Zhang, Dimitris Samaras, Dardo Tomasi, Nelly Alia-Klein, Lisa Cottone, Andreana Leskovjan, Nora Volkow, and Rita Goldstein. Exploiting temporal information in functional magnetic resonanace imaging brain data. In *Medical Image Computing and*

Computer-Assisted Intervention, pages 679–687. Springer Berlin / Heidelberg, 2005. 1.3