Hidden Process Models

Rebecca A. Hutchinson

May 26, 2006

Abstract

Functional Magnetic Resonance Imaging (fMRI) technology allows us to study the brain in action at the resolution of millimeters and seconds. Machine learning provides a principled approach to studying the data collected through fMRI in hopes of better understanding cognitive behavior. Our goal is to use machine learning to improve the cognitive modeling process by connecting hypotheses about processes occurring in the brain during particular tasks to fMRI data collected while subjects perform those tasks. This would permit hypotheses to be compared, modified, and even learned based on experimental data.

In this thesis, we propose a class of probabilistic models designed for fMRI data called Hidden Process Models (HPMs). HPMs assume a system of partially observed, linearly additive processes that overlap in space and time, motivated by characteristics of cognitive processes. In preliminary experiments, HPMs have shown promise in analyzing both real and synthetic fMRI data, but they need some extensions to be widely useful. For instance, we plan to extend the parameterization of HPMs, improve the inference algorithm for HPMs, and develop algorithms for learning HPMs under more types of uncertainty.

1 Introduction

Functional Magnetic Resonance Imaging (fMRI) is a safe, non-invasive technology that allows us to collect measurements of brain activity at a temporal resolution on the order of seconds and a spatial resolution on the order of millimeters. We can use fMRI to study the brain by designing experiments in which subjects perform controlled tasks in the scanner. fMRI has the potential to revolutionize the way that cognitive theories are developed and evaluated, because it affords us the opportunity to compare cognitive models to data from real human subjects.

Machine learning techniques provide a principled approach to analyzing fMRI data. Two significant challenges to doing machine learning in this domain are that

fMRI data is high-dimensional and sparse. In a typical experiment, the brain is divided into about 10,000 volume elements, or voxels, each of which is imaged approximately every second for perhaps 15-20 minutes. Over this time, we usually only acquire a few dozen examples of the tasks that the subject performs. This results in a very large feature set (voxels at time points), for which we may have only 10-40 training trials from which to learn.

Another challenge we face in analyzing fMRI data is due to the nature of the fMRI signal: it is a highly noisy measurement of an indirect and temporally blurred neural correlate. fMRI measures changes in the blood oxygenation level (also called the *hemodynamic response* or the *Blood Oxygen Level Dependent (BOLD) response*). The hemodynamic response to a short burst of less than a second of neural activity lasts for 10-12 seconds. The fMRI signal we acquire then is really a series of temporally and/or spatially overlapped hemodynamic responses which represent neural activity smeared over time.

Our goal in applying machine learning to fMRI data is to improve the cognitive modeling process. We envision a setting in which a cognitive scientist can compare several hypotheses of the cognitive processes a subject invokes during a particular task based on the likelihood of the experimental data under each model. Consider for example a study in which subjects in the scanner repeatedly view a picture and read a sentence and indicate whether the sentence correctly describes the picture. We would like a method in which the cognitive scientist can compare a simple model in which the only two cognitive processes are ReadSentence and ViewPicture to a slightly more complex model adding a third process, Decide, or to even more complex models of this task. A convincing justification for a theory would be its ability to outperform competing theories in explaining real human fMRI data.

To create such a tool for cognitive scientists, we must overcome the challenges of learning in the fMRI domain and provide a model framework powerful enough to deal with uncertainty about the processes underlying the data. In short, our problem is to learn the parameters and timing of potentially overlapping, partially observed responses to cognitive processes in the brain using many features and a small number of noisy training examples.

2 Related Work

The most common approach to modeling fMRI data in the neuroimaging community is to employ multiple regression methods based on the General Linear Model (GLM) (e.g., [7], [8]). In these methods it is assumed that the fMRI data is generated by a collection of cognitive processes with *known* timing and identities. The activation at each voxel at each time is assumed to be governed by a Gaussian

distribution whose mean is the linear sum of contributions from all processes active at that time. Each process is assumed to generate a spatio-temporal signature in the observed fMRI data, where the process' contribution to each voxel at each time relative to the process onset may be characterized by an independent parameter. Given known timings and identities of a set of processes, and a sequence of observed fMRI data, multiple regression methods can be used to find maximum likelihood estimates of the spatio-temporal signature of each process. While this GLM approach captures modeling assumptions which have been found very useful for fMRI analysis, it is restricted to the case where process timings and identities are known.

Another approach to modeling time series data, which has not been widely used for fMRI analysis, is Dynamic Bayesian Networks (DBNs) [19, 11]. For instance, we could model fMRI data with factorial Hidden Markov Models (fHMMs) [13, 12], in which each hidden Markov chain represents a cognitive process that can be on or off. However, applying fHMMs is problematic for our task because the hemodynamic response renders fMRI data non-Markovian, and because the fHMM is unable to represent multiple instantiations of the same cognitive process overlapping in time without significant computational and modeling overhead. Furthermore, the sample complexity of learning unconstrained DBNs will often be prohibitive given the sparsity of available fMRI data. We require a more constrained, informed model to learn in the presence of such sparse training data.

Other tools for analyzing fMRI data include ACT-R [1] and 4-CAPS [15]. Based on production rules, these systems make predictions about the fMRI data that will be observed when subjects in the scanner do algebra problems or read sentences, respectively. This approach revolutionized cognitive science by providing a framework in which cognitive theories can be specified via computer programs whose output can be compared to real fMRI data. We would like to provide a method in which cognitive theories can be partially specified and the details can be filled in by learning them from real data. That is, we would like to develop a data-driven model, rather than taking a predictive approach.

In approaching the problem of tracking cognitive processes in the brain using fMRI data, it is instructive to note that several machine learning techniques have been successful in analyzing fMRI data for slightly different purposes. For example, several groups have been able to accurately classify non-overlapping windows of fMRI data. For instance, [18] trained classifiers for differentiating between windows where the subject was viewing a picture or reading a sentence (as in the experiment alluded to above), for differentiating different types of sentences, and for differentiating 12 semantic categories of stimuli. Other groups have also trained classifiers to tell apart several semantic categories of stimuli, such as photographs of faces versus chairs [14]. Classifiers such as linear discriminant analysis and

support vector machines have even been successful at classifying different stimuli categories across multiple sessions for the same subject [6]. These methods do not directly address the problem of tracking unknown cognitive processes, but it is encouraging to note that enough useful information is captured in the fMRI signal to do these classification tasks, despite the noise level and sparsity of the data.

While all of these approaches have been important and useful to the fMRI community, none of them quite achieve the goal we have outlined for fMRI data analysis. The GLM provides a method for estimating the hemodynamic response from the data, but requires us to assume that we know the timing of all the cognitive processes in advance, which is overly restrictive. DBNs are a natural choices for time series modeling, but we have concerns about the sample complexity issues we face with such sparse data. ACT-R and 4-CAPS can make predictions about fMRI data for some tasks, but we would like to learn cognitive models from the data. We would like a method that combines the best of these techniques; the model should learn process hemodynamic responses from real data with minimal assumptions about their timing, track cognitive processes over time, and provide cognitive scientists a framework for expressing theories and validating them with fMRI data.

3 Thesis

The central thesis of this work is that we can develop machine learning techniques to simultaneously estimate a hemodynamic response *and* the timing of the cognitive processes that generate it. We have designed a method specifically for this purpose called Hidden Process Models (HPMs). HPMs are a potentially significant contribution to the field because existing methods require either the timing or the hemodynamic response of a set of events to be assumed in advance in order to analyze the data. HPMs on the other hand allow researchers to search for hidden cognitive events that do not correspond directly to stimuli, and to compare competing theories in a principled manner driven by experimental data.

Hidden Process Models are a probabilistic model for multivariate time series data generated by a system of partially observed, linearly additive processes that overlap in space and time. We use HPMs to model fMRI data by assuming there is a partially observed series of hidden, overlapping cognitive processes in the brain that probabilistically generate the observed fMRI time series. HPMs represent an intermediate point between GLM approaches and DBNs on the spectrum of expressibility versus learnability. HPMs extend the expressiveness of the GLM by allowing us to learn hemodynamic responses for processes whose timing is unknown. They are a more constrained approach than DBNs in that they incorporate

more domain knowledge than general DBNs, which improves their learnability. More details on how HPMs relate to the GLM and to DBNs are given in Appendices A and B, respectively.

Our work so far on HPMs has shown that they can accomplish the goal of simultaneously estimating the hemodynamic response and timing of events in constrained situations. For instance, HPMs can currently estimate nonparametric hemodynamic responses for datasets in which we know the number of processes and the durations of their responses in advance, and we have a relatively small number of timing combinations to consider. Much of the future work in this area focuses on removing these types of constraints so that HPMs can work in more general problem situations. For instance, we would like to allow parametric hemodynamic responses, more timing possibilities, variable durations, and unknown numbers of processes. We will discuss these extensions in more detail in Section 5.

4 Preliminary Work

This section describes our work so far on Hidden Process Models. To date, we have established a formal notation for HPMs, developed inference and learning algorithms for HPMs, and applied these algorithms to real and synthetic fMRI data.

In order to describe HPMs more clearly, we introduce a particular dataset to use as a concrete example. This dataset was obtained from an fMRI study [16] in which human subjects were presented a sequence of 40 trials. In half of these trials subjects were presented a picture (involving the symbols *, +, and $) for 4 sec followed by a blank screen for 4 sec, followed by a sentence (e.g. “The star is above the plus.”). They then pressed a button to indicate whether the sentence correctly described the picture. In the other half of the trials the sentence was presented first and the picture second, using the same timing. Throughout the session, fMRI images of brain activity were captured every 500 msec (i.e., TR = 500 msec). In the real data, each image was summarized in terms of the mean activation in 7 pre-defined regions. We also created synthetic data to match the timing of this experiment, in which we varied experimental parameters like the number of voxels and the signal-to-noise ratio. Our goal in applying HPMs to this data is to model the underlying cognitive processes used by subjects to perform their task.

4.1 HPM Formalism

Figure 1 depicts an HPM for the synthetic picture/sentence data. HPMs assume the observed time series data is generated by a collection of hidden process instance.

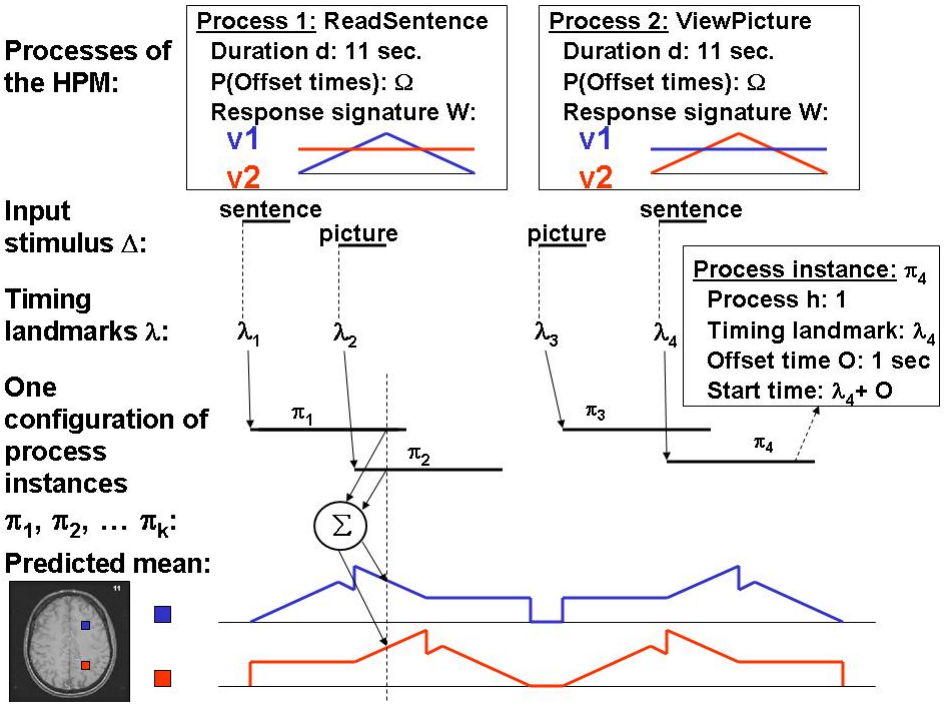

Figure 1: Synthetic 2-process HPM. Processes each have a duration, timing distribution, and response signature over time and space. They are instantiated multiple times, at start times depending on timing landmarks derived from the input stimuli and the timing distribution of the process. The predicted mean of the data is the sum of the contributions of each active process instance at each time point.

Each process instance is active during some time interval, and influences the observed data only during this interval. Process instances inherit properties from general process descriptions. The timing of process instances depends on timing parameters of the general process it instantiates, plus a fixed timing landmark derived from input stimuli. 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.

More formally, we consider the problem setting in which we are given observed data $\mathbf{Y}$ and known input stimuli $\mathbf{\Delta}$. The observed data $\mathbf{Y}$ is 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 locations in the brain. The information about input stimuli, $\mathbf{\Delta}$, is a $T \times I$ matrix, where matrix element $\delta_{ti} = 1$ if an input stimulus of type i is initiated at time t, and $\delta_{ti} = 0$ otherwise. The observed data Y is generated nondeterministically by some system in response to the input stimuli $\mathbf{\Delta}$. We use an HPM to model this system. Let us begin by defining processes:

Definition. A *process* h is a tuple $\langle \mathbf{W}, \Theta, \Omega, d \rangle$. d is a scalar called the *duration* of h, which specifies the length of the interval during which h is active. $\mathbf{W}$ is a $d \times V$ matrix called the *response signature* of h, which specifies the influence of h 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 the distribution over a discrete-valued random variable which governs the timing of h, and which takes on values from Ω. The set of all processes is denoted by $\mathcal{H}$.

We will use the notation $\Omega(h)$ to refer to the Ω for a particular process h. 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 hypothesize general cognitive processes such as ReadSentence, ViewPicture, and Decide, each of which is instantiated once for each trial. The instantiation of a process at a particular time is called a *process instance*, defined as follows:

Definition. A *process instance* π is a tuple $\langle h, \lambda, O \rangle$, where h identifies a *process* as defined above, λ is a known scalar called a *timing landmark*, and O is an integer random variable called the *offset time*, which takes on values in $\Omega(h)$. The time at which process instance π begins is defined to be $\lambda + O$. The multinomial distribution governing O is defined by $\Theta(h)$. The duration of π is given by $d(h)$, and the response signature is $W(h)$.

The timing landmark λ is defined by a particular input in $\mathbf{\Delta}$ (e.g., the timing landmark for a ReadSentence process instance may be the time at which the sentence stimulus is presented to the subject), whereas the values for the offset time O and/or the process h of the process instance may in general be unknown. We model the distribution over O as a property of the process, and its particular value as a property of the instance; that is, while there may be slight variation in the offset times of ReadSentence instances, we assume that in general the amount of time between a sentence stimulus and the beginning of the ReadSentence cognitive process follows the same distribution for each instance of the ReadSentence process.

The latent variables in an HPM are h and O for each of the process instances. We refer to each possible set of process instances as a *configuration*.

Definition. A *configuration* c is a set of fully-specified process instances $\{\pi_1 \ldots \pi_L\}$.

Given a configuration $c = \{\pi_1 \ldots \pi_L\}$ the probability distribution over each observed data point y_{tv} in the observed data $\mathbf{Y}$ is defined by the Normal distribution:

$$y_{tv} \sim \mathcal{N}(\mu_{tv}(c), \sigma_v) \tag{1}$$

where σ_v is the standard deviation characterizing the time-independent noise distribution associated with the v^{th} time series, and where

$$\mu_{tv}(c) = \sum_{\pi \in c} \sum_{\tau=0}^{d(h(\pi))} \delta(\lambda(\pi) + O(\pi) = t - \tau)\, w_{\tau v}(h(\pi)) \tag{2}$$

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

Equation (2) says that the mean of the Normal distribution governing observed data point y_{tv} is the sum of single contributions from each process instance whose interval of activation includes time t. In particular, the $\delta(\cdot)$ expression is non-zero only when the start time ($\lambda(\pi)+O(\pi)$) of process instance π is exactly τ time steps before t, in which case we add the element of the response signature $\mathbf{W}(h(\pi))$ 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 [5] and is widely used in fMRI data analysis.

We can now define Hidden Process Models:

Definition. A *Hidden Process Model*, *HPM*, is a tuple $\langle \mathcal{H}, \Phi, \mathcal{C}, \langle \sigma_1 \ldots \sigma_V \rangle \rangle$, where $\mathcal{H}$ is a set of processes, Φ is a vector of parameters defining the prior probabilities over the processes in $\mathcal{H}$, $\mathcal{C}$ is a set of candidate *configurations*, and σ_v is the standard deviation characterizing the noise in the v^{th} time series of $\mathbf{Y}$.

Note that the set of configurations C is defined as part of the HPM. Each configuration is an assignment of timings and process types to some number of process instances. This restricts the hypothesis space of the model, and facilitates the incorporation of timing constraints as mentioned above (e.g. if none of the configurations allow process instance n to be of type ReadSentence and/or start at $t = 4$, then that possibility is not considered by the HPM).

An *HPM* defines a probability distribution over the observed data $\mathbf{Y}$, given input stimuli $\mathbf{\Delta}$, as follows:

$$P(\mathbf{Y}|\mathit{HPM}, \mathbf{\Delta}) = \sum_{c \in \mathcal{C}} P(\mathbf{Y}|\mathit{HPM}, C = c) P(C = c|\mathit{HPM}, \mathbf{\Delta}) \tag{3}$$

where $\mathcal{C}$ is the set of candidate configurations associated with the *HPM*, and C is a random variable defined over $\mathcal{C}$. Notice the term $P(\mathbf{Y}|\mathit{HPM}, C = c)$ is defined by equations (1) and (2) above. The second term is

$$P(C = c|\mathit{HPM}, \mathbf{\Delta}) = \frac{\prod_{\pi \in c} P(h(\pi)|\mathit{HPM}) P(O(\pi)|h(\pi), \mathit{HPM}, \mathbf{\Delta})}{\sum_{c' \in \mathcal{C}} \prod_{\pi' \in c'} P(h(\pi')|\mathit{HPM}) P(O(\pi')|h(\pi'), \mathit{HPM}, \mathbf{\Delta})} \tag{4}$$

where $P(h(\pi)|\mathit{HPM})$ is the prior probability of process $h(\pi)$ as defined by the parameter vector Φ of the *HPM*. Similarly, $P(O(\pi)|h(\pi), \mathit{HPM}, \mathbf{\Delta})$ is the multinomial distribution defined by $\Theta(h(\pi))$.

Thus, the generative model for an *HPM* involves first choosing a configuration $c \in \mathcal{C}$, using the distribution given by equation (4), then generating values for each time series point using the configuration c of process instances and the distribution for $P(\mathbf{Y}|\mathit{HPM}, C = c)$ given by equations (1) and (2).

4.2 Inference with HPMs

The basic inference problem in HPMs is to infer the posterior distribution over the candidate configurations $\mathcal{C}$ of process instances, given the *HPM*, input stimuli $\mathbf{\Delta}$, and observed data $\mathbf{Y}$. By Bayes theorem we have

$$P(C = c|\mathbf{Y}, \mathbf{\Delta}, \mathit{HPM}) = \frac{P(\mathbf{Y}|C = c, \mathit{HPM}) P(C = c|\mathbf{\Delta}, \mathit{HPM})}{\sum_{c' \in \mathcal{C}} P(\mathbf{Y}|C = c', \mathit{HPM}) P(C = c'|\mathbf{\Delta}, \mathit{HPM})} \tag{5}$$

where the terms in this expression can be obtained using equations (1), (2), and (4).

4.3 Learning HPM Parameters

The learning problem in HPMs is: given an observed data sequence $\mathbf{Y}$, an observed stimulus sequence $\mathbf{\Delta}$, and a set of candidate configurations including landmarks for each process instance, we wish to learn maximum likelihood estimates of the HPM parameters. The set Ψ of parameters to be learned include $\Theta(h)$ and $\mathbf{W}(h)$ for each process $h \in \mathcal{H}$, Φ, and σ_v for each time series v.

4.3.1 Learning from fully observed data

First consider the case in which the configuration of process instances is fully observed in advance (i.e., all process instances, including their offset times and processes, are known). For example, in our sentence-picture brain imaging experiment, if we assume there are only two cognitive processes, ReadSentence and ViewPicture, then we can reasonably assume a ReadSentence process instance begins at exactly the time when the sentence is presented to the subject, and ViewPicture begins exactly when the picture is presented.

In such fully observable settings the problem of learning Φ and the $\Theta(h)$ reduces to a simple maximum likelihood estimate of multinomial parameters from observed data. The problem of learning the response signatures $\mathbf{W}(h)$ is more complex, because the $\mathbf{W}(h)$ terms from multiple process instances jointly influence the observed data at each time point (see equation (2)). Solving for $\mathbf{W}(h)$ reduces to solving a multiple linear regression problem to find a least squares solution, after which it is easy to find the maximum likelihood solution for the σ_v. Our multiple linear regression approach in this case is based on the GLM approach described in [7]. 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.

4.3.2 Learning from partially observed data

In the more general case, the configuration of process instances may not be fully observed, and we face a problem of learning from incomplete data. In this section we consider the case where the offset times of process instances are unobserved, however the number of process instances is known, along with the process associated with each. For example, in the sentence-picture brain imaging experiment, if we assume there are three cognitive processes, ReadSentence, ViewPicture, and Decide, then while it is reasonable to assume known offset times for ReadSentence and ViewPicture, we must treat the offset time for Decide as unobserved.

In this case, we use an EM algorithm to obtain locally maximum likelihood estimates of the parameters, based on the following Q function. Here we use C to denote the collection of unobserved variables in the configuration of process instances, and we suppress mention of $\mathbf{\Delta}$ to simplify notation.

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

The EM algorithm finds parameters Ψ that locally maximize the Q function by iterating the following steps until convergence:

E step: Solve for the probability distribution over the unobserved features of configurations of process instances. The solution to this is given by equation (5).

M step: Use the distribution over configurations from the E step to obtain parameter estimates that maximize the expected log likelihood of the full (observed and unobserved) data.

The update to $\mathbf{W}$ is the solution to a weighted least squares problem minimizing the objective function

$$\sum_{v=1}^{V}\sum_{t=1}^{T}\sum_{c\in\mathcal{C}} -\frac{P(C=c|\mathbf{Y},\Psi^{\text{old}})}{2\sigma_v^2}\ \left(y_{tv}-\mu_{tv}(c)\right)^2 \tag{6}$$

where $\mu_{tv}(c)$ is defined in terms of W as given in equation (2).

The updates to the remaining parameters are given by

$$\sigma_v \longleftarrow \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_{tv}^2 - 2y_{tv}E_{C|\mathbf{Y},\Psi^{\text{old}}}[\mu_{tv}(C)] + E_{C|\mathbf{Y},\Psi^{\text{old}}}[\mu_{tv}^2(C)]\right)}$$

$$\theta_{h,O=o} \longleftarrow \frac{\sum_{c\in\mathcal{C}}\sum_{\pi\in c}\delta(h(\pi)=h)\delta(O(\pi)=o)P(C=c|\mathbf{Y},\Psi^{\text{old}})}{\sum_{c\in\mathcal{C}}\sum_{\pi\in c}\delta(h(\pi)=h)\sum_{o'\in\Omega(h(\pi))}\delta(O(\pi)=o')P(C=c|\mathbf{Y},\Psi^{\text{old}})}$$

4.3.3 Model selection

In cases where the exact number of processes or the identities of the processes are not known in advance, we can use cross-validated likelihood to choose the most appropriate model from a set of candidate HPMs.

4.4 Experimental Results

To test the effectiveness of the HPM learning and inference algorithms, we applied them to both synthetic data and to fMRI data obtained from human subjects. Experiments with fMRI data were used to elucidate the hidden cognitive processes in human subjects, and test HPMs on problems of realistic complexity. Experiments with synthetic data allowed us to measure the effect of noise, number of training examples and data dimensionality on the ability to accurately learn HPMs.

4.4.1 Results on fMRI Data

We experimented with three different HPMs to analyze the fMRI data described above:

1. **HPM-2**: An HPM with two processes, ReadSentence and ViewPicture, each with a specified duration of 11 seconds (to account for the hemodynamic response), and where the onset of each process is specified in advance to coincide exactly with the appearance of the corresponding stimulus. Thus, the timing is fully specified, and the only HPM parameters to be learned are the response signatures for the two processes.

2. **GNB**: An HPM with two processes, identical to HPM-2 except that durations of both processes were set to 8 seconds (the time between stimuli) instead of 11. This models the ReadSentence and ViewPicture processes without overlap. The generative model learned by this HPM is equivalent to the generative model learned by a Gaussian Naive Bayes (GNB) classifier where the classes are ReadSentence and ViewPicture, and the examples to be classified are 8-second windows of fMRI observations.

3. **HPM-3**: An HPM with three processes: ReadSentence, ViewPicture, and Decide, each with a duration of 11 seconds. The timings for ReadSentence and ViewPicture were fully specified, but the onset of the Decide process was not. Instead, we assigned a uniform prior to start times in the interval beginning with the second stimulus and ending 5 seconds later. The model was constrained to assume that the onset of the Decide process, while unknown, was at the same point in each of the 40 trials.

For each of five human subjects, we trained and evaluated these three HPMs, using a 40-fold leave-one-trial-out cross validation method. Data likelihood was measured over the left-out trials. While the training process allowed some variation in process instance timings as mentioned above, the instances' process types were known. We also measured the accuracy of the HPMs in classifying the identities of the first and second process instances in each left-out trial (i.e., classifying ReadSentence versus ViewPicture). The classification was performed by choosing the process with highest posterior probability given the observed data and the learned HPM, marginalizing over the possible process identities for the remaining process.

The results are summarized in Table 1. Note first that both HPM-2 and HPM-3 outperformed GNB in both data log likelihood and classification accuracy. The comparison between GNB and HPM-2 is especially noteworthy because the only

difference between these two models is the 8 second duration (resulting in non-overlapping processes) versus 11 seconds. Essentially, HPM-2 classifies the data interval by simultaneously deconvolving the contributions of the two overlapping processes, and assigning the classes (process identities), whereas the standard GNB classifier is unable to model the overlap. HPM-3 goes even further than HPM-2, by assuming the existence of a third process with unknown onset time, and simultaneously estimating the contributions of each of these three, together with assigning process identities. We take these results as a promising sign of the superiority of HPMs over earlier classifier methods (e.g.,[18]) for modeling cognitive processes.

Second, notice that HPM-3 outperforms HPM-2. This indicates that HPMs provide a viable approach to modeling truly hidden cognitive processes (e.g., the Decide process) with unknown timing. The fact that the 3-process model has greater cross-validated data log likelihood supports the hypothesis that subjects are invoking three processes rather than two when performing this task. While the existence of the Decide process may be intuitively obvious, the point is that HPMs offer a principled basis for resolving questions about the number and nature of hidden and overlapping cognitive processes. The learned response signatures of HPM-3 for one subject are shown in Figure 2.

Finally, we applied HPMs to a second fMRI study in which subjects were presented a sequence of 120 words, one every 3-4 seconds, and pressed a button to indicate whether the word was a noun or verb. In this study, images were obtained once per second (i.e., TR = 1 sec). We trained a two-process HPM, with processes ReadNoun and ReadVerb, each with duration 15 seconds. This implies overlapping contributions from up to 5 distinct process instances in the observed fMRI data at any given time, making it unrealistic to apply classifiers like GNB to this data. We applied learned HPMs to classify which process instances were ReadNoun versus ReadVerb. Despite the overlapped responses, we found cross-validated classification accuracies significantly (p-value < 0.1) better than random classification in 4 of 6 human subjects, with the accuracy for the best subject reaching .67 (random classification yields accuracy of .5). This further supports our claim that HPMs provide an effective approach to analyzing overlapping cognitive processes in realistic fMRI experimental datasets.

4.4.2 Results on Synthetic Data

The synthetic data shown in Figure 3 is for one voxel generated by an HPM containing three processes. The timing of the processes was selected to mimic the real dataset described above. Roughly, the triangle-shaped process corresponds to the ReadSentence process, the square-shaped process is View Picture, and the parabola-shaped process is Decide. We can generate synthetic datasets (in which

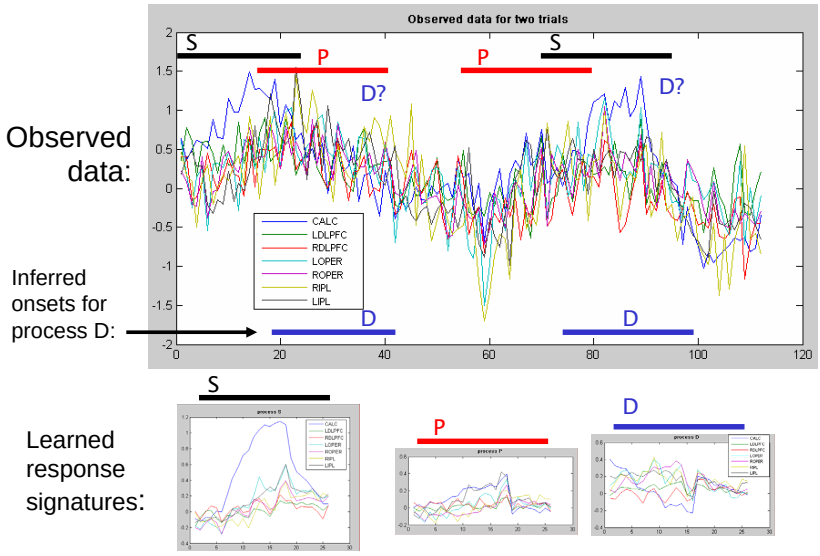

Figure 2: Learned HPM-3 process responses for one subject: fMRI data. The top plot shows two trials. The bottom plots are learned response signatures for ReadSentence (S), ViewPicture (P), and Decide (D). Each line represents data from one of the 7 brain regions.

we know ground truth about the processes underlying the data) in which we vary the number of voxels, amount of noise, the number of training trials, and the proportion of voxels containing relevant signal.

We trained HPMs on these datasets with constraints on process IDs and timings that would be reasonable assumptions for corresponding real data. For instance, the first two process instances in a trial were ReadSentence or View Picture in either order, but repetitions like ReadSentence followed by ReadSentence were not allowed. Timings were constrained to be close to the stimuli with offsets of only 0 or 1 . The third process instance in each trial was known to be Decide, but the offset from the second stimulus varied from 0 to 5. Each HPM was used to choose process IDs and timings on an independent test set with the same number of voxels

Table 1: fMRI study: leave-one-trial-out cross validation results for GNB, HPM-2, and HPM-3 on the five subjects (A through E) exhibiting the highest accuracies (acc) and data log likelihoods (loglik) out of 13 total subjects. The accuracies are for predicting the identities of the first and the second stimuli (up to 80 correct answers, 0.5 for purely random classification scheme).

	A	B	C	D	E
GNB acc	0.725	0.750	0.725	0.637	0.750
HPM-2 acc	0.750	0.875	0.700	0.675	0.787
HPM-3 acc	0.775	0.875	0.738	0.637	0.812
GNB loglik	-896	-786	-941	-783	-476
HPM-2 loglik	-876	-751	-912	-768	-466
HPM-3 loglik	-864	-713	-898	-753	-447

and amount of noise as the training set.

We can use these synthetic datasets to investigate many questions about HPMs. For example, we can ask how inference scales with the numbers of voxels and training trials. To answer this question, we generated synthetic data as described above, with $\sigma_v = 1$ for all voxels. We used the HPM that generated the data to do inference over test sets of 40, 80, 120, and 160 trials, and let the number of voxels range from 500 to 10,000. The results are shown in Figure 4, and indicate that inference scales linearly with the number of voxels and the number of trials. Each trial had the same structure, with 48 possible configurations fitting the assumptions described above, but the HPM does not exploit this structure, so these results should generalize to experiments with more varied trial types.

5 Proposed Work

While HPMs have already shown promise in fMRI analyses, there is room for improvement. Recall that the goal of HPMs is to improve the cognitive modeling process by connecting hypotheses of cognitive behavior to empirical fMRI data in a principled way. Toward this goal, we would like HPMs to handle larger and more complex problems, to support commonly used assumptions in the fMRI data analysis community, and to be as efficient as possible. We see three main research thrusts that work toward these aims: the parameterization of the model, dealing with timing constraints, and learning under different kinds of uncertainty. We discuss these areas in more detail below.

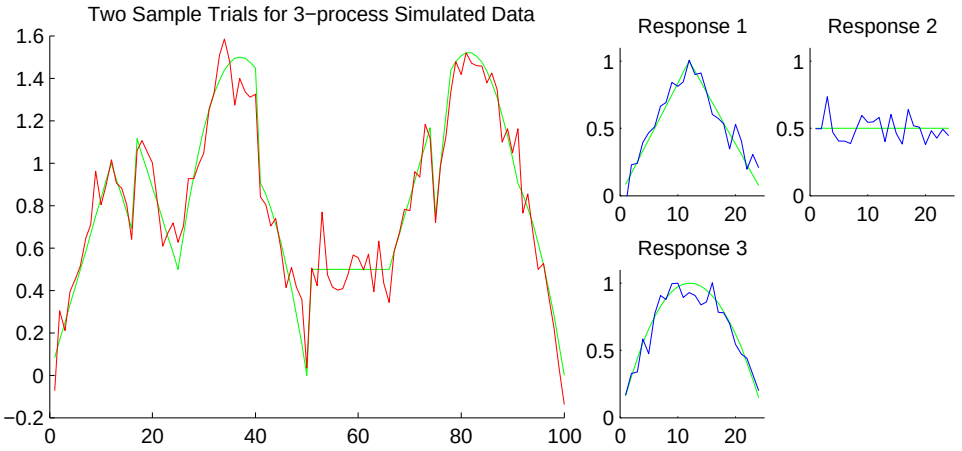

Figure 3: Learned versus true process responses: synthetic data. Plots on the right show learned response signatures (blue lines) for three processes superimposed on the true response signatures (green lines). This HPM was learned from the synthesized data shown on the left, in red; the green line indicates the synthesized data before noise was added.

In addition to improving the HPM formalism and algorithms, we would like to identify an open question in the field of cognitive science on which HPMs can shed new light. As we have discussed, HPMs provide an opportunity to ask a new kind of question about fMRI data: What does the hemodynamic response of a hidden process (i.e. not perfectly correlated with stimuli) look like? We are currently engaged in a literature search to find the right domain to illustrate the power of HPMs. Such a domain should have processes with a temporal resolution of several seconds; simple stimuli like flashing checkerboards are processed too quickly to be interesting for HPMs. Additionally, the domain should involve some task that does not directly correspond to understanding stimuli so that some set of one or more hidden processes is a reasonable modeling assumption. Some of the domains we are considering are language processing, decision making, emotion, and pain.

5.1 Model Parameterization

One area of HPMs that merits further investigation is the way in which we parameterize the model. Our main goals in improving the parameterization of HPMs are to reduce the sample complexity of the learning problem and to support commonly

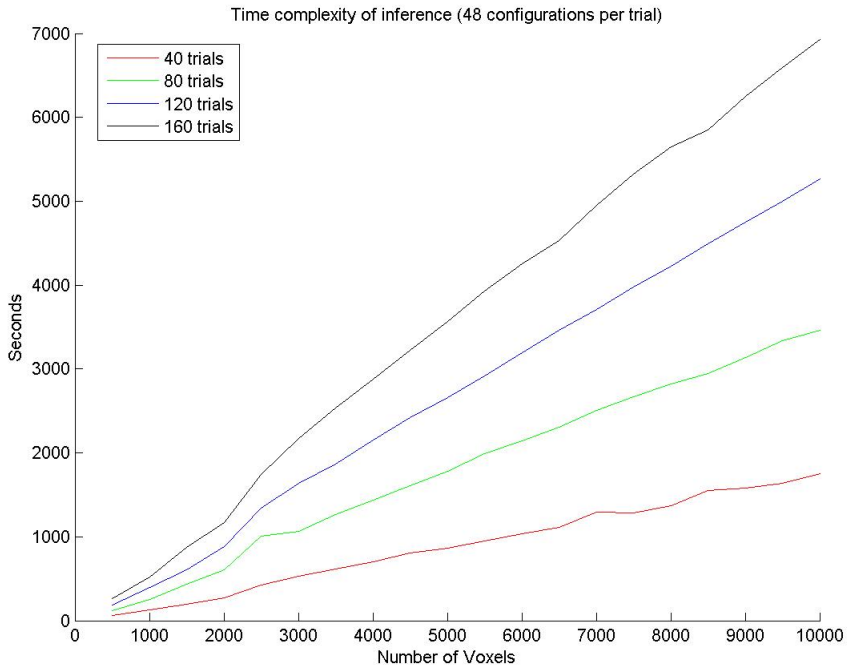

Figure 4: Empirical analysis of scaling properties of inference in HPMs. Inference appears to scale linearly with the number of voxels and the number of trials.

used assumptions in the fMRI data analysis community. Specifically, we want to look into sharing model parameters, using parametric forms and/or smoothing for the process response signatures, and allowing characteristics of the stimuli to affect response signatures.

5.1.1 Parameter Sharing

Sharing HPM parameters could potentially reduce the sample complexity of our learning problem, allowing us to make better use of the sparse data we have. In [20], Niculescu implemented one form of parameter sharing for HPMs in which voxels share the shape of their response signatures. In that work, the response signatures are modeled with a set of parameters (nonparametric weights, as in the current version of HPMs) describing a canonical signature for a region of the brain, along with scaling parameters for each voxel in that region. In the given region,

this reduces the number of parameters used to model a response signature from $duration(process) * V$ to $duration(process) + V$. This framework was implemented under the assumption that the process instances timings were fully observed; we would like to extend this work to the case where the timings are only partially observed. To extend this work to the case of unknown timings, we must reformulate the objective function (Equation 6) of the reweighted least squares optimization problem from the M step in terms of the new parameters (the scaling factors and canonical response). We are currently studying convex optimization to learn how to do this [4].

5.1.2 Parametric Response Signatures

The current version of Hidden Process Models takes a nonparametric approach to the shape of the process response signatures by modeling them with d weights, where d is the duration of the process. This allows the response to have any shape, and effectively samples that shape at the temporal resolution of the experiment. This approach has worked well so far, and will continue to be a good choice in some situations. In other situations, we may wish to parameterize the response signatures differently. The benefits of supporting parametric response signatures in HPMs are threefold: we may be able to reduce the number of parameters to be estimated in learning HPMs, we may be able to use prior knowledge about response shapes to better fit the data, and we can support a variety of modeling assumptions made in fMRI data analysis, making HPMs more widely applicable.

In [5], Boynton uses a parametric form for the impulse response function, which is the response to a short, simple stimulus (flashing checkerboards). In that work, a gamma function with 2 free parameters is used to model the impulse response function, and the estimated response is used to investigate the linear systems approach to fMRI analysis. While hemodynamic responses to more complex stimuli could deviate significantly from impulse responses to short simple stimuli, it would be interesting to see whether a gamma-shaped hemodynamic response could be used to model more complex cognitive tasks. We expect there to be a bias-variance trade-off between a simple but biased response shape with few parameters versus a nonparametric unbiased response shape with many parameters (like the current model). Another point on the spectrum of this bias-variance trade-off might be the spline models discussed in [10]. The first technical challenge to incorporating either parametric form into HPMs is to figure out how to deconvolve the responses and estimate their parameters, first for known timing and then for unknown.

Another way to model the hemodynamic response function would be to use hemodynamic basis functions as mentioned in [7]. This approach is quite similar

to the reweighted least squares procedure we currently use to estimate the non-parametric form of the response signatures, except that the responses are projected into a lower-dimensional subspace so that we may estimate fewer parameters. Dale gives equations for doing maximum likelihood estimation in this case. One significant challenge to this approach is choosing the right subspace (and corresponding basis functions). We are also studying basis function regression in [2] for the details of learning in this setting.

5.1.3 Smoothed Response Signatures

In cases where we choose not to commit to a particular parameterization of the hemodynamic response, we might still want to impose a smoothness constraint on the learned weights. Smoothing might help us deal with noise on the data, and may yield more accurate, biologically plausible response estimates. Intuitively, a simple constraint might be that the difference between adjacent weights is upper-bounded by some parameter Δ. This constraint could be implemented as a regularizer added to the objective function of the reweighted least squares procedure in the M step. The strength of the regularizer could be moderated by a smoothing parameter, chosen by cross-validation. Again, we are studying convex optimization for the details [4].

We could also use insight from the fMRI literature to derive a smoothing constraint for the weight parameters. For instance, previous work [5] suggests that the hemodynamic response is made up of a sequence of impulse responses to short stimuli. These impulse responses are more commonly accepted to be gamma-shaped than longer, more complex hemodynamic responses to higher-level cognitive events. Thus, while we might not want to commit to a gamma-shaped response signature, perhaps we could bound the decay of the hemodynamic response based on the decay of a gamma-shaped impulse response and use this in our regularizer instead.

5.1.4 Stimulus-Dependent Response Signatures

Another way in which we would like to extend the HPM parameterization is to allow characteristics of the stimuli to affect the process response signatures. For instance, in experiments where subjects are reading sentences, we might want the length, type, or truth value of the sentence to affect the response in some way. More complex sentences might elicit additional brain activity.

Note that another way to capture some of the effects of stimuli on process responses would be to split the process into two separate processes (e.g. ReadSimpleSentence and ReadComplexSentence). There are a few reasons why it might

be advantageous to use something like a scaling parameter instead. One is that we may have prior knowledge about the relationship between the two responses, which can inform our model. Another is that we may be able to model the two responses with fewer parameters if the relationship between them is simple. Of course, if the effect of a stimulus parameter is so complex that it requires more parameters to describe it than to estimate a whole second process response signature, we should model the second response separately.

5.2 Inference and Learning Under Timing Constraints

Perhaps the biggest weakness of the current version of HPMs is the way that timing constraints (like "process instances of type A begin at some t offset in $\{0\ 1\ 2\}$ seconds after their corresponding stimulus λ") are incorporated into the model. While we believe the timing constraints to be an important tool for analyzing fMRI data, the way that we currently specify them is inefficient. Right now, the HPM itself includes a set of process configurations that describe the allowable timings of the process instances. Timing constraints are observed by simply not putting any configurations into the model that violate the constraints, essentially limiting the hypothesis space of the model to be consistent with the timing constraints. This makes the inference procedure easy (try each configuration and pick the one that maximizes the data likelihood) but it is inefficient to list all possible configurations, much less to evaluate them all. The enumeration of these configurations also requires a large design matrix to be created for the reweighted least squares M step. Our goals here are to make our algorithms as efficient as possible so we can handle larger problems and to make any existing limitations clear to the cognitive modeler in terms of fMRI experiment design so that HPMs can be used to their fullest advantage.

5.2.1 Current Limitations

We believe that the first step toward improving this aspect of HPMs is to formally specify the impact of the current scheme on our performance. We would like to know exactly how enumerating the configurations limits us. Answering this question will involve analytical analysis and synthetic data experiments looking into the number of parameters to estimate, the complexities of the inference and learning algorithms, and the size of the problems HPMs can deal with. For instance, we know that the size of the design matrix exceeds Matlab memory limitations in the sentence/picture dataset for 40 trials and 2000 voxels under the timing constraints described above, but we would like a more formal specification of the size of this matrix in terms of the HPM parameters. We saw above that the inference algorithm

scales linearly with voxels and trials, but we need to look at how it scales with the number of configurations per trial. We also expect the notion of identifiability to play a part in this question. Given a rigorous analysis of the current method, we can use this information in one of two ways.

5.2.2 Possible Solutions for Inefficient Inference

The first way we see to deal with the inefficiencies resulting from enumerating the configurations is to translate the limitations into the language of fMRI experiment design. This approach accepts the shortcomings of HPMs and focuses instead on understanding how to work within the limits of the method. The idea behind this solution is that even if HPMs are limited to, for instance, a 20-minute fMRI experiment with 4 cognitive processes or fewer (which actually covers many fMRI experiments), they still represent a novel, useful way to study the brain, and it would be helpful to provide researchers with guidelines for designing fMRI experiments for HPM analyses. While this approach is not a perfect solution, it still makes a valuable contribution to the fMRI data analysis community and is almost certainly possible to complete as part of this thesis.

The second solution we see is to develop more efficient exact and/or approximate algorithms for HPMs. This approach is desirable in that it is likely to make HPMs applicable to more problems, but it is much more difficult and uncertain than the first solution. We have many questions about how we might attempt this, but few answers. Some of the research areas we will look to for insight on this problem include Markov Chain Monte Carlo sampling methods [17] and DBN algorithms. One approach would be to implement something like Gibbs sampling over configurations, but this would only improve the E step; the memory issue in the M step would be unchanged. It may be necessary to move away from configurations as a tool for expressing timing constraints and address the latent variables O and h of the process instances in some other way.

5.3 Learning Under Uncertainty

A third direction we see for improving HPMs is to develop learning algorithms allowing more kinds of uncertainty in the training data. The current algorithms work for fully observed training data, and for training data in which the process timings and/or the process IDs are partially observed. However, these algorithms assume a known number of processes in the model, and known response signature durations for those processes. Our goals in allowing more types of uncertainty in learning are to ease the cognitive modeling process by requiring fewer parameters to be set by the cognitive scientist and to let the data inform the model as much as

possible.

5.3.1 Uncertainty in Process Durations

One type of uncertainty we want to deal with is the length of the hemodynamic response. The current model allows different processes to have different durations, but all the durations are assumed to be known in advance. Instead, we would like to automatically learn the appropriate duration for each process. A naive, inefficient approach would be to generate and test several HPMs with varying durations for the processes. Perhaps an alternative heuristic would be to start with arbitrarily long durations and set a threshold for the weights. At regular intervals, we could check whether the last weight in the response is above the threshold; if not, we remove the last weight from the response and check the next one. The threshold for lowering the process duration could be based on some estimate of the noise in the data. For example, if the last weight in a process response has been learned to be 0.1, and we estimate the σ for that voxel to be 0.5, we decrement the process duration and model the time points immediately following this process as noise rather than an effect of the process itself.

Note that in the nonparametric version of HPMs, we do not suffer much by assuming known durations because the model is free to learn trailing zeros in the response weights. On the other hand, if we model process response signatures using hemodynamic basis functions, we can in some sense learn the process durations for free by using basis functions of varying lengths. Each process response signature would then consist of weights on the basis functions that would determine the duration of the process.

5.3.2 Uncertainty in the Number of Processes in the Model

Another type of uncertainty we face is about the number of processes underlying the system. Right now, we address this uncertainty by doing model selection; for instance, we have trained HPMs with 2 and 3 processes on the same data, and used data log-likelihood to choose the more appropriate model. We would like to investigate the possibility of learning the number of processes underlying the model from data during training. This problem seems to have some parallels to nested Chinese Restaurant Processes [3]; perhaps that literature could inspire a solution. We expect this task to be quite difficult, and we will have to decide how important it is to our goal of improving cognitive modeling. To compare a small number of relatively simple models, using cross-validation to do model selection is feasible and has the advantage of giving a good quality measure for comparison (the cross-validated accuracy or likelihood). Choosing the number of processes

automatically becomes more important as the number and complexity of models increases, but it is not clear whether the fMRI domain will challenge the limits of the cross-validation scheme yet.

5.4 More Ideas

The three research areas listed above will constitute the main thrusts of this thesis. The final thesis will delve into some subset (not necessarily all) of the specific questions in each research area. However, we are not at a loss for research projects extending HPMs; here are several other research ideas we would be interested in approaching, time permitting.

While the linear systems approach to fMRI data analysis is widely used, significant deviations from linearity have also been shown in fMRI [9]. Our model will suffer from nonlinearities in the fMRI signal. Two significant nonlinear effects are habituation and saturation. The idea of habituation is that the amplitude of hemodynamic responses to rapidly repeated stimuli decreases as the brain adjusts to the demands of the task. HPMs assume that the response to each of the rapid stimuli is the same and do not account for the subject becoming accustomed to doing the task. The idea of saturation is that the sum of the amplitudes of many simultaneously occurring hemodynamic responses is bounded by the vascular structure of the brain region. HPMs allow an arbitrary number of hemodynamic responses to be piled atop one another, summing their weights to predict an unrealistically high fMRI signal. An interesting extension to HPMs would be to model habituation and/or saturation in some way.

We might like to apply HPMs to another domain. While they have been developed for fMRI data analysis, we believe the approach could be useful for other time series domains. Additionally, applying HPMs to different kinds of datasets will likely inspire more ideas for extensions of the model that arise in response to the needs of a different domain.

6 Schedule

Below is a rough schedule of the proposed work to be done on this thesis, organized by publication deadlines.

NeuroImage Journal (Summer 2006): Extended paper reporting progress on HPMs so far to the neuroscience community. This should include the treatment of HPMs given in this document, progress on model parameterization, Niculescu's work on parameter sharing [20], and a version of the first solution to our inefficient inference problem (specifying limits in terms of experiment design).

AAAI/ICML (February 2007): Paper reporting progress on inference and learning algorithms.

NIPS (June 2007): Paper applying HPMs to an open question in the field of cognitive science.

Projected completion: December 2007.

7 Conclusion

Hidden Process Models (HPMs) provide a general formalism for representing probability distributions over time series data. Here we have described the formalism and associated inference and learning methods, and presented experimental results showing the ability of these algorithms to learn HPMs characterizing hidden cognitive processes in human subjects while their brain activity is recorded in an fMRI scanner. We have also outlined an agenda for future research on HPMs, in which the three main research areas are the parameterization of the model, dealing with timing constraints, and learning under uncertainty. In relating HPMs to other approaches, we have found that HPMs provide an intermediate point between GLM regression and DBNs on the spectrum of expressivity versus learnability.

A HPMs and the GLM

HPMs are related to the General Linear Model (GLM) which is widely used for fMRI data analysis in the neuroscience community. HPMs provide a key generalization of the standard GLM multiple regression methods used for fMRI analysis because HPMs allow uncertainty regarding the timings of the hidden processes, whereas standard GLM regression analyses (e.g., [7]) assume the precise timings of each process are known in advance.

To show the correspondence between HPMs and the GLM more precisely, we follow the overview of GLMs for fMRI analysis from [7]. Consider the case where we have just one voxel whose observed discrete time series is given by the column vector $\mathbf{y}$ of dimension T. The GLM models this time series as

$$\mathbf{y} = \mathbf{X_1 w_1} + \mathbf{X_2 w_2} + \ldots + \mathbf{X_K w_K} + \mathbf{n} \tag{7}$$

where $\mathbf{w_k}$ is a discrete-time vector of dimension M representing the hemodynamic response function (in our terms, the response signature) associated with the k^{th} process. Here $\mathbf{X_k}$ is the $T \times M$ dimensional binary matrix which represents the exact timing of all instantiations of the k^{th} process, where the value of $\mathbf{X_k}(q, r)$ is 1 if an instance of process k was initiated at time $q - r + 1$, and 0 otherwise. The

T dimensional vector $\mathbf{n}$ represents a vector of zero mean Gaussian noise which is temporally uncorrelated.

We can represent equation (7) in matrix form, and also generalize it to the case where there are multiple voxels, yielding the matrix equation

$$\mathbf{Y} = \mathbf{XW} + \mathbf{N} \tag{8}$$

where $\mathbf{Y}$ is the horizontal concatenation of the observed time series vectors for the different voxels, $\mathbf{X} = [\mathbf{X_1} \cdots \mathbf{X_K}]$ is the horizontal concatenation of the timing matrices for the K processes, $\mathbf{W}$ is the vertical concatenation of the response matrices for the processes, and $\mathbf{N}$ is the horizontal concatenation of the noise vectors for the different voxels.

Equation (8) corresponds to the special case of an HPM model where the HPM *configuration* (i.e., all process timings and process identities) is given in advance. In this case, $\mathbf{Y}$ and $\mathbf{X}$ are both known, and we need only solve for the response signatures of the processes, represented by $\mathbf{W}$. The maximum likelihood solution for $\mathbf{W}$ can be obtained using Ordinary Least Squares methods. HPMs generalize the problem setting by treating the timing matrix $\mathbf{X}$ as *unknown*; that is, treating $\mathbf{X}$ as a random variable to be estimated (subject to constraints derived from prior knowledge) simultaneously with $\mathbf{W}$. Given the widespread use and success of the more restricted GLM regression model in fMRI analysis, the generalization provided by HPMs has many potential applications in this domain.

B HPMs and DBNs

HPMs correspond to a constrained subclass of Dynamic Bayes Nets that make the following additional modeling assumptions:

1. Events are modeled at the granularity of process instances with start times and durations rather than state values at every time point.

2. Parameter sharing is enforced between all instances of the same process.

3. The HPM learning algorithm easily accommodates constraints of the form "Process A occurs n times within the interval [t,t']."

To see the correspondence between HPMs and DBNs, it is instructive to encode these three types of constraints using DBNs. A natural starting point is a factorial Hidden Markov Model (fHMM) [13]. Each hidden Markov chain (i.e., each hidden state variable in the fHMM) can represent a process, and instances of this process

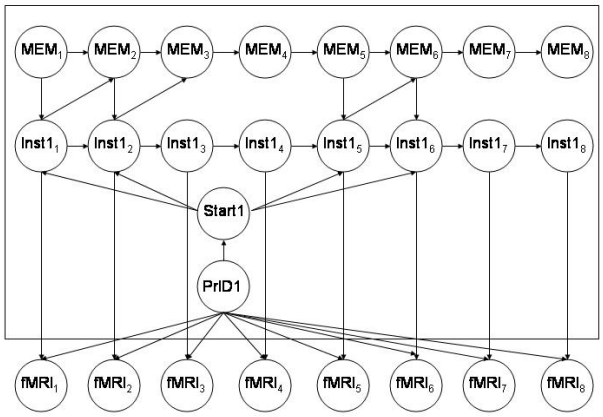

Figure 5: Part of a DBN capturing the same assumptions and constraints as HPMs. The variables in the box must be repeated for each process instance. In this case, we know that Inst1 occurs once on t=[1,8]. This DBN reflects a timing model in which the process type probabilistically influences the possible start times for its instances, so depending on Inst1's process type, it might start at t=$\{$1,2,5,6$\}$. When it starts, Inst1 will go from 0 to its duration and count back down to 0. MEM is needed to ensure Inst1 occurs *exactly* once; if the duration is 3 and the process starts at t=1, Inst1 must not restart at t=5 or t=6 even though its value has returned to 0, and if the process did not start at t=1 or t=2 or t=5, it must start at t=6.

can be represented by letting the state variable take on integer values from 0 (indicating the process is inactive) up to the maximum duration of the process. The variable can take on its maximum value whenever its process is instantiated, and count down on each transition, returning to zero when the process instance terminates. Although this use of fHMMs successfully captures the assumption that a process occurs over some fixed interval, we cannot represent overlapping instances of the same process in this fashion. If we try to encode overlapping process instances by summing the state values each would produce individually, we are unable to uniquely decompose this sum into the multiple process instance timings that produced it. To allow overlapping instances of the same process, the Markov chains in the model must represent *process instances* rather than processes. In this case each chain can have a static variable to indicate its process type, and we can then enforce sharing of parameters between the process instances with like types. Note

if we take this approach, the effective number of parameters needed to fully define each Markov chain is just two: the time at which the process instance begins, and the type of process it instantiates. HPMs represent each process instance using exactly these two parameters. This approach requires that we assume the number of process instances in the model in advance; this assumption is also made in the HPM formalism given above.

Property 3 is the most difficult property to embed in the DBN model. The constraint "Process A is instantiated once during the interval $[t,t']$" means that there is a finite subset of possible start times for this instance of process A. We need additional variables to keep track of which start times are allowed for instance A, plus a memory chain to keep track of whether or not the instance has already begun. This memory is necessary if the possible start times span more than the duration of the process so that a second instance does not occur on the same interval. A DBN that is equivalent to an HPM is shown in Figure 5. This DBN has no more free parameters than its equivalent HPM. Its conditional probability distributions can be filled in with the HPM parameters discussed above, plus some deterministic tables for counting and memory.

As this example illustrates, encoding an HPM within the generic DBN framework is possible, but not elegant. Of course in either formalism, encoding the domain knowledge of Properties 1-3 will reduce the effective number of hidden parameters to be estimated, and will also improve the learnability of the model. HPMs provide a convenient, process-oriented formalism to represent and work within these modeling assumptions.

References

[1] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004. http://act-r.psy.cmu.edu/about/.

[2] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] D. Blei, T. Gri, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process, 2004.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] 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.

[6] 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.

[7] Anders M. Dale. Optimal experimental design for event-related fMRI. *Human Brain Mapping*, 8:109–114, 1999.

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

[9] K. J. Friston, A. Mechelli, R. Turner, and C. J. Price. Nonlinear responses in fMRI: The balloon model, volterra kernels, and other hemodynamics. *NeuroImage*, 12:466–477, 2000.

[10] C. Genovese. A bayesian time-course model for functional magnetic resonance imaging, 2000.

[11] Zoubin Ghahramani. Learning dynamic Bayesian networks. *Lecture Notes in Computer Science*, 1387:168–197, 1998.

[12] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Proc. Conf. Advances in Neural Information Processing Systems, NIPS*, volume 8, pages 472–478, 1995.

[13] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.

[14] 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.

[15] Marcel Adam Just, Patricia A. Carpenter, and Sashank Varma. Computational modeling of high-level cognition and brain function. *Human Brain Mapping*, 8:128–136, 1999. http://www.ccbi.cmu.edu/project_10modeling4CAPS.htm.

[16] T.A. Keller, M.A. Just, and V.A. Stenger. Reading span and the time-course of cortical activation in sentence-picture verification. In *Annual Convention of the Psychonomic Society*, 2001.

[17] D.J.C. MacKay. Introduction to monte carlo methods. In Michael I. Jordan, editor, *Learning In Graphical Models*, pages 175–204. Kluwer Academic Publishers, 1998.

[18] Tom M. Mitchell et al. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

[19] Kevin P. Murphy. Dynamic bayesian networks. To appear in *Probabilistic Graphical Models*, M. Jordan, November 2002.

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