Hidden Process Models

Rebecca A. Hutchinson RAH@CS.CMU.EDU
Tom M. Mitchell TOM.MITCHELL@CMU.EDU
Indrayana Rustandi INDRA@CS.CMU.EDU
Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 USA

Abstract

We introduce Hidden Process Models (HPMs), a class of probabilistic models for multivariate time series data. The design of HPMs has been motivated by the challenges of modeling hidden cognitive processes in the brain, given functional Magnetic Resonance Imaging (fMRI) data. fMRI data is sparse, high-dimensional, non-Markovian, and often involves prior knowledge of the form "hidden event A occurs n times within the interval $[t,t']$." HPMs provide a generalization of the widely used General Linear Model approaches to fMRI analysis, and HPMs can also be viewed as a subclass of Dynamic Bayes Networks.

1. Introduction

We introduce the Hidden Process Model (HPM), a probabilistic model for multivariate time series data. HPMs assume the data is generated by a system of partially observed, linearly additive processes that overlap in space and time. While we present a general formalism for any domain with similar modeling assumptions, HPMs are motivated by our interest in studying cognitive processes in the brain, given a time series of functional magnetic resonance imaging (fMRI) data. We use HPMs to model fMRI data by assuming there is an unobserved series of hidden, overlapping cognitive processes in the brain that probabilistically generate the observed fMRI time series.

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. It is natural to think of the observed fMRI sequence as arising from a set of hidden cognitive processes in the subject's brain, which we would like to track. To do this, we use HPMs to learn the probabilistic time series response signature for each type of cognitive process, and to estimate the onset time of each instantiated cognitive process occurring throughout the experiment.

There are significant challenges to this learning task in the fMRI domain. The first is that fMRI data is high-dimensional and sparse. A typical fMRI image measures a correlate of neural activity at a resolution of a few millimeters, providing an image with approximately 10,000 voxels (three dimensional pixels). Images are typically collected once per second in experiments typically lasting 15-20 minutes. Experiments often involve dozens of repeated trials during which similar stimuli are presented. 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. A second challenge 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*). The hemodynamic response to a short burst of less than a second of neural activity lasts for 10-12 seconds. This temporal blurring in fMRI makes it problematic to model the time series as a first-order Markov process. 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.

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., (Dale, 1999)). 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. HPMs provide a generalization of this GLM approach to cover the case where process timings and identities are not known in advance.

A second approach to modeling time series data, which has not been widely used for fMRI analysis, is Dynamic Bayesian Networks (DBNs) (Murphy, 2002; Ghahramani, 1998). HPMs provide a formalism which is more con-

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.

strained (e.g. HPMs do not allow arbitrary transitions among hidden states) than general DBNs. In fact, HPMs can be mapped into DBNs without requiring any new free parameters, and thus can be considered a constrained subclass of DBNs that reduce the sample complexity for learning by embedding additional assumptions.

Like GLM-based approaches, and like DBNs, HPMs assume the observed multivariate time series is characterized by some set of latent variables. The key modeling assumptions made by HPM are:

- HPMs model the latent time series as a set of processes, each of which endures for some time interval. This is in contrast to unconstrained DBNs which would allow arbitrary hidden state transitions rather than enforcing a boxcar shape on the timing.

- Process instances refer to general descriptions of process types. Many of the parameters of the process instance (e.g., constraints on its timing, its spatio-temporal signature in the data) are inherited from these general process descriptions. For example, a general ReadSentence process might be instantiated many times during an experiment, but each instance shares the same signature.

- HPMs easily encode prior knowledge of the form "process instance X occurs somewhere inside the time interval [a,b]." For example, we know that an instance of the ReadSentence process occurs sometime between the sentence stimulus presentation and the subject's response. We give this information to the HPM by restricting its hypothesis space to configurations that are consistent with this constraint, rather than allowing it to consider explanations of the data in which, for example, the ReadSentence process begins before the sentence stimulus.

2. Formalism

HPMs assume the observed time series data is generated by a collection of hidden process instances, as depicted in Figure 1. 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 $\boldsymbol{\Delta}$.

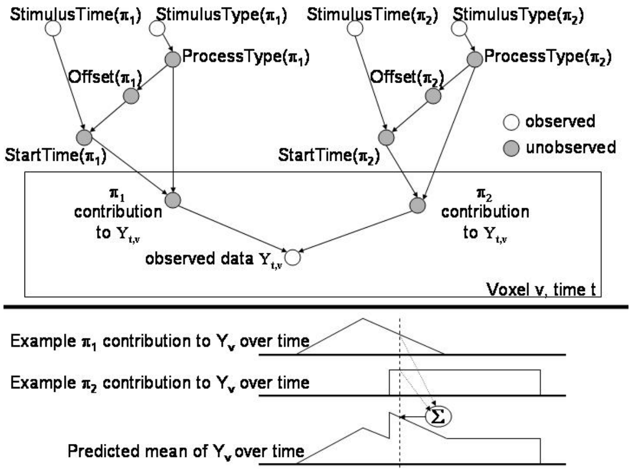

Figure 1. **Top:** A Hidden Process Model drawn as a graphical model. The observed data $Y_v t$ for voxel v at time t depends on contributions from some set of process instances (in this case π_1 and π_2). The contribution from a process instance depends on its start time and its process type. The process type references a general process description which specifies a response signature, duration, and timing distribution for its instances. The process type and the start time of a process instance depend on input stimuli. **Bottom:** Hidden Process Models assume that the contributions of process instances sum linearly to produce the mean of a normal distribution governing $Y_v t$. In this example, the response signature of the process type for π_1 in voxel v is a triangle shape, and the response signature of the process type for π_2 in voxel v is a square. These response shapes are placed in time according to the start times for π_1 and π_2 and summed to obtain the predicted mean in voxel v.

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, $\boldsymbol{\Delta}$, 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 $\boldsymbol{\Delta}$. 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 one of our fMRI studies (see Section 4) subjects had to determine whether a sentence correctly described a picture, on each of 40 trials. 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 $d(h)$.

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 process 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 process instances $\{\pi_1 \ldots \pi_L\}$ with their parameters ($\{\lambda, O, d\}$) fully-specified.

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:

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

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

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

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 (Boynton et al., 1996) 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 conditional probabilities over the processes in $\mathcal{H}$ given the stimulus types in $\mathbf{\Delta}$, $\mathcal{C}$ 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:

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

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 $P(\mathbf{Y}|\textit{HPM}, C = c)$ is defined by equations (1) and (2) above. The second term is

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

where $P(h(\pi)|\textit{HPM}, \mathbf{\Delta})$ is the conditional probability of process $h(\pi)$ given the stimuli indicated by $\mathbf{\Delta}$ as defined by the parameter vector Φ of the *HPM*. Similarly,

$P(O(\pi)|h(\pi), \mathit{HPM}, \boldsymbol{\Delta})$ 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).

3. Algorithms

3.1. Inference

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 $\boldsymbol{\Delta}$, and observed data $\mathbf{Y}$. By Bayes theorem we have

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

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

3.2. Learning

The learning problem in HPMs is: given an observed data sequence $\mathbf{Y}$, an observed stimulus sequence $\boldsymbol{\Delta}$, 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.

3.2.1. 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 (Dale, 1999). 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.

3.2.2. 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 Φ, 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 $\boldsymbol{\Delta}$ 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 the process instances 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 maximizing the objective function

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

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

The updates to the remaining parameters are given by

$$\sigma_v \longleftarrow \sqrt{\frac{1}{T}\sum_{t=1}^{T} a_t}$$

where

$$a_t = \left(y_{tv}^2 - 2y_{tv}E_{C|\mathbf{Y},\Psi^{\text{old}}}[\mu_{tv}(C)] + E_{C|\mathbf{Y},\Psi^{\text{old}}}[\mu_{tv}^2(C)]\right)$$

and

$$\theta_{h,O=o} \longleftarrow \frac{\sum_{c\in\mathcal{C}}\sum_{\pi\in c}\delta(h(\pi)=h)b_{c,\pi,o}}{\sum_{c\in\mathcal{C}}\sum_{\pi\in c}\delta(h(\pi)=h)\sum_{o'\in\Omega(h(\pi))}b_{c,\pi,o'}}$$

where

$$b_{c,\pi,o} = \delta(O(\pi)=o)P(C=c|\mathbf{Y},\Psi^{\text{old}})$$

4. Experiments

We applied HPMs to both synthetic and real fMRI data. The real dataset arises from the fMRI study mentioned earlier, in which subjects view a picture and sentence, and must decide whether the sentence correctly describes the picture. The synthetic data was constructed to match the timing of the real dataset, but with simplified response signatures.

4.1. Experiments with Synthetic Data

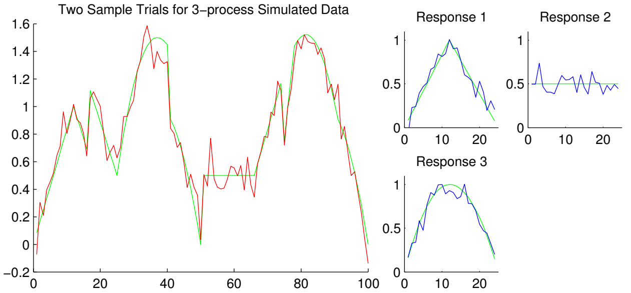

Figure 2. Learned versus true process responses: synthetic data. Plots on the right show learned response signatures (blue lines) for three processes superimposed on the true response signatures (green lines). This HPM was learned from synthesized data the example on the left, in red; the green line indicates the synthesized data before noise was added.

The synthetic data shown in Figure 2 was generated by an HPM containing three processes. During training, the exact timing was given in advance for the first two processes, but not for the third. The HPM learning algorithm obtained good estimates of the true response signatures despite strong overlaps in the time intervals of the processes instances and significant noise in the data. In a variety of experiments we measured the accuracy of learned HPMs by the fit of their response signatures to true response signatures, by their data log likelihood on held out data, and by their ability to correctly classify the process associated with each process instance on held out data. Accuracy decreased with increasing data noise and improved with the number of trials in the time series. We also found accuracy improved as the dimension of the data increased, presumably because the larger dimension provides more information for localizing the hidden timing of process instances. In this synthetic dataset, all voxels contained relevant signal for each process instance. We are working on new datasets that contain irrelevant voxels in addition to the informative ones to more closely model real fMRI data.

4.2. Experiments with Real Data

The fMRI data used in this experiment was obtained from an fMRI study (Keller et al., 2001) in which human subjects were presented a sequence of 40 trials. In half of these trials subjects were presented a picture for 4 sec followed by a blank screen for 4 sec, followed by a sentence. 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). Each image was summarized in terms of the mean activation in 7 pre-defined regions.

Our goal in applying HPMs to this data is to model the underlying cognitive processes used by subjects to perform their task. We experimented with three different HPMs to analyze this data:

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 thirteen 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.,(Mitchell et al., 2004)) 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. (Note that we must use cross validation to avoid overfitting and favoring the mor complex model.) The learned response signatures of HPM-3 for one subject are shown in Figure 3.

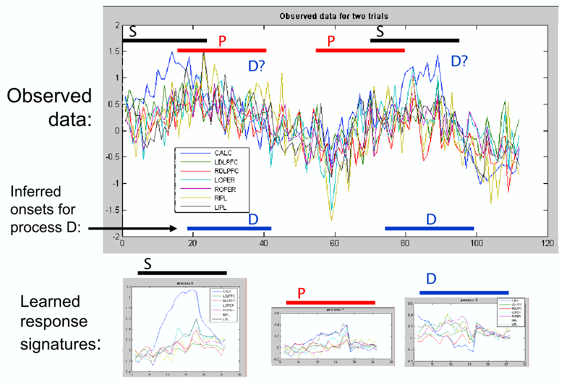

Figure 3. Learned HPM-3 process responses for one subject: fMRI data. The top plot shows two trials. The bottom plots are learned response signatures for ReadSentence (S), ViewPicture (P), and Decide (D). Each line represents data from one of the 7 brain regions.

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.

Table 1. fMRI study: leave-one-trial-out cross validation results for GNB, HPM-2, and HPM-3 on five subjects (A through E) exhibiting the highest accuracies (top 3 rows) and data log likelihoods (bottom 3 rows) out of 13 total subjects, and the average over all 13 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).

	A	B	C	D	E	Avg
GNB	0.725	0.750	0.725	0.637	0.750	0.610
HPM-2	0.750	0.875	0.700	0.675	0.787	0.630
HPM-3	0.775	0.875	0.738	0.637	0.812	0.660
GNB	-896	-786	-941	-783	-476	-840
HPM-2	-876	-751	-912	-768	-466	-819
HPM-3	-864	-713	-898	-753	-447	-811

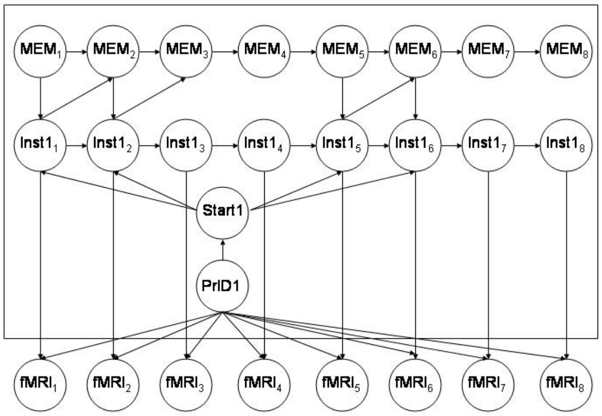

Figure 4. Example of a partial DBN capturing the same assumptions and constraints as HPMs. The variables in the box must be repeated for each process instance (e.g. $Inst1_t$). In this example, we know that $Inst1$ occurs exactly once on $t = [1, 8]$. Suppose further that the process type $PrID1$ limits the possible start times to $t = \{1, 2, 5, 6\}$. $Inst1$ is an integer-valued random variable; when it starts, its value is set to its duration and it counts 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 $t = 6$. This DBN has no more free parameters than its corresponding HPM, but is not an elegant description of our modeling assumptions.

5. Related Work

5.1. HPMs and DBNs

We mentioned earlier that HPMs correspond to a constrained subclass of Dynamic Bayes Nets that make additional modeling assumptions. To encode these modeling assumptions in a DBN, we can use integer-valued Markov chains to model process instances. The chains can count down deterministically from their duration to zero to indicate the interval during which the instance is active. The Markov chains must be process instances instead of processes to allow overlapping instances of the same process. Chains can be linked to process ID variables to keep track of their process types and associated parameters. Additional variables are required to keep track of which start times are allowed for the process instance Markov chains. Furthermore, we need a memory chain for each process instance to keep track of whether it has started yet. These variables are explained in more detail through the example in Figure 4. As this example illustrates, encoding an HPM within the generic DBN framework with no new free parameters is possible, but not elegant. Of course in either formalism, encoding the additional domain knowledge captured by our modeling assumptions 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.

DBNs have also been used for other purposes in fMRI analysis, as in (Højen-Sørensen et al., 2000) and (Zhang et al., 2006).

5.2. HPMs and the General Linear Model

HPMs are also 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., (Dale, 1999)) assume the precise timings of each process are known in advance. The GLM models the time series with the following equation:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{N} \tag{7}$$

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 (7) 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.

6. Discussion

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. HPMs provide an intermediate point between GLM regression and DBNs on the spectrum of expressivity versus learnability.

HPMs address a key open question in fMRI analysis: how can one learn the response signatures of overlapping cognitive processes with unknown timing? There is no competing method to HPMs available in the fMRI community, and general DBNs will not suffice because they do not constrain the learning problem sufficiently to allow learning from sparse fMRI data sets. We see many directions for future work on HPMs.

Perhaps the most significant limitation of the current version of HPMs is the way that timing constraints (like “process instances of type A begin at some t offset [0...2] seconds after their corresponding stimulus λ”) are incorporated into the model. Currently, the HPM 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. We are looking into approximate inference algorithms for HPMs in an effort to make them suitable for larger problems.

Another area in which HPMs can be improved is to reduce the number of independent parameters estimated for HPM response signatures, perhaps using parametric forms such as those in (Boynton et al., 1996), or parameter sharing as in (Niculescu & Mitchell, 2006). Another approach would be to incorporate hierarchical Bayes approaches that allow learning priors on the parameter values by pooling across different voxels or processes within a subject, or by pooling across different subjects.

Finally, we see HPMs as a framework that may be useful in other high-dimensional, sparse-data domains where generic DBNs are not sufficiently constrained to be of use. For example, in problems such as tracking people in a building given a distributed network of sensors, it may be that modeling assumptions of HPMs can be used to model processes such a “person walking down the hallway,” and incorporating prior knowledge such as “this person walks down the hallway exactly once during time interval [t1,t2].”

References

Boynton, G. M., Engel, S. A., Glover, G. H., & Heeger, D. J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1. *The Journal of Neuroscience*, *16*, 4207–4221.

Dale, A. M. (1999). Optimal experimental design for event-related fMRI. *Human Brain Mapping*, *8*, 109–114.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks. *Lecture Notes in Computer Science*, *1387*, 168–197.

Højen-Sørensen, P., Hansen, L. K., & Rasmussen, C. E. (2000). Bayesian modelling of fMRI time series. *Proc. Conf. Advances in Neural Information Processing Systems, NIPS* (pp. 754–760).

Keller, T., Just, M., & Stenger, V. (2001). Reading span and the time-course of cortical activation in sentence-picture verification. *Annual Convention of the Psychonomic Society*.

Mitchell, T. M., et al. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, *57*, 145–175.

Murphy, K. P. (2002). Dynamic bayesian networks. To appear in *Probabilistic Graphical Models*, M. Jordan.

Niculescu, R., & Mitchell, T. (2006). Bayesian network learning with parameter constraints. *Journal of Machine Learning Research*.

Zhang, L., Samaras, D., Alia-Klein, N., Volkow, N., & Goldstein, R. (2006). Modeling neuronal interactivity using dynamic bayesian networks. In Y. Weiss, B. Schölkopf and J. Platt (Eds.), *Advances in neural information processing systems 18*, 1595–1602. Cambridge, MA: MIT Press.