Hidden Process Models

Tom M. Mitchell[1,2,3], Rebecca Hutchinson[2], Indrayana Rustandi[2,3]
{tom.mitchell,rah,indra}@cs.cmu.edu

February 17, 2006

CMU-CALD-05-116

Center for Automated Learning and Discovery[1]
Computer Science Department[2]
Center for the Neural Basis of Cognition[3]
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We introduce the Hidden Process Model (HPM), a probabilistic model for multivariate time series data intended to model complex, poorly understood, overlapping and linearly additive processes. HPMs are motivated by our interest in modeling cognitive processes given brain image data. We define HPMs, present inference and learning algorithms, study their characteristics using synthetic data, and demonstrate their use for tracking human cognitive processes using fMRI data.

1 Introduction

In this paper, we propose the Hidden Process Model (HPM), a probabilistic model for multivariate time series data generated by a system of overlapping, potentially hidden, linearly additive processes. HPMs are motivated by the study of cognitive processes in the brain using functional magnetic resonance imaging (fMRI) data, a technique to indirectly capture neural activations in a subject's brain by measuring changes in the blood oxygenation level (also called the *hemodynamic response*). In particular, HPMs are designed to learn and track both known and hidden cognitive processes, taking into account that the hemodynamic response signatures might overlap in the fMRI data.

HPMs build on existing machine learning methods for time series data and the state-of-the-art approach for fMRI data analysis. With respect to the former, HPMs have similarities to dynamic Bayesian networks (DBNs) [1]. In fact, we have found that HPMs can be expressed in DBN format, and thus are technically a special case of DBNs. However, to preserve the set of assumptions captured in the HPM format requires a complex DBN. For instance, we must inflate the state-space of the DBN by using Markov chains as binary 'memory' variables. We are continuing work on formalizing the connection between HPMs and DBNs, but at this point we suspect that HPMs will provide an advantage over their DBN counterparts in terms of time and sample complexities.

With respect to fMRI data analysis, HPMs build on a variant of the General Linear Model (GLM) approach widely used in fMRI data analysis. In particular, HPMs are similar to the GLM approach described in [3] to extract hemodynamic responses out of overlapping processes. Our work differs from theirs in that HPMs can handle processes with unknown timing, whereas GLMs do not allow uncertainly about timing in the design matrix. HPMs express that uncertainty probabilistically, where every instance of a general process shares the same timing distribution. Although one could attempt to handle timing uncertainly by enumerating and solving a set of alternative GLMs, HPMs provide a more principled way to describe timing uncertainty, and a principled method for learning process models in the face of this uncertainty.

There has been an effort to analyze fMRI data using hidden Markov models (HMMs) [4]. Unlike that approach, HPMs are not restricted to block design fMRI data and are capable of inferring states that are not binary.

2 Hidden Process Models

Informally, 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 processes are simultaneously active at some 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

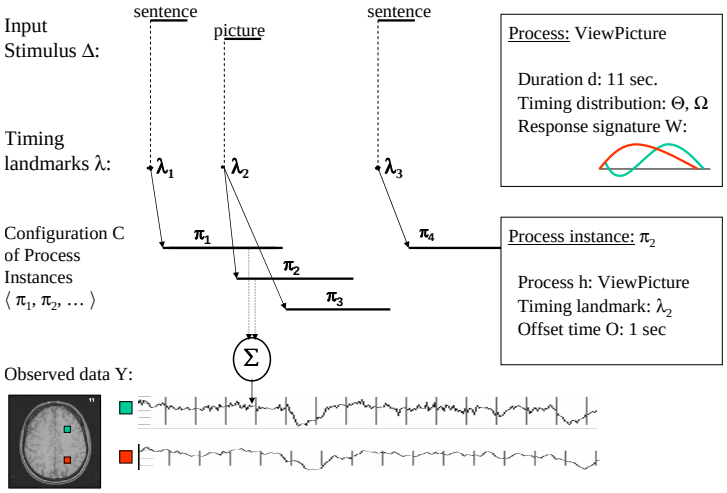

Figure 1: Hidden Process Models assume data is generated by a collection of process instances that inherit properties from general process descriptions.

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 the collection of parameters for a multinomial distribution of a random variable which governs the timing of h, and which takes on values in 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 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 is 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.

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 $\{\pi_1 \ldots \pi_L\}$.

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 $\mathbf{Y}$.

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).

2.1 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 $\mathbf{\Delta}$, 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).

2.2 Learning

The learning problem in HPMs is analogous to that for HMMs and DBNs: 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.

2.2.1 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 Θ_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 approach described in [3]. 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.

2.2.2 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, 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: The E step involves solving for the probability distribution over the unobserved features of configuration of process instances. The solution to this is given by our earlier equation (5).

M step: The M step uses the distribution over the partially observed 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

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

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

The updates to the remaining parameters are given by

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

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

2.2.3 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.

2.3 Tractability and prior knowledge

HPMs can be mapped into fHMMs by creating a fHMM state variable for each HPM process, and defining the appropriate fHMM emission distribution. The advantage of HPMs is that their different timing model naturally incorporates prior

assumptions that yield large reductions in the number of latent variables to be estimated. Given an HPM with L processes and M process instances and an observed time series of length T, unconstrained fHMMs would require consideration of 2^{LT} configurations of state variables, whereas HPMs consider only "LT choose M" configurations. Further reductions follow when one has prior knowledge of which process is associated with each process instance (reducing the number of configurations to fewer than T^M). Large additional reductions occur when the time series can be partitioned into segments separated by intervals with zero process instances (as is common in brain imaging experiments with rest periods between trials). For example, in an experiment involving n trials with maximum trial length τ and m process instances per trial, the number of configurations considered reduces to $n\tau^m$.

3 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 synthetic data allowed us to measure the effect of noise, number of training examples and data dimensionality on the ability to accurately learn HPMs. Experiments with fMRI data were used to elucidate the hidden cognitive processes in human subjects, and test HPMs on problems of realistic complexity.

3.1 Experiments with synthetic data

Data was synthesized from a known HPM with three processes whose response signatures are shown in Figure 2. Data was synthesized to mimic the characteristics of the fMRI data set discussed in the following section: the data series consisted of a sequence of trials, each trial instantiating all three processes. During learning, the exact timing for two processes was provided, but not for the third. As shown in the figure, the HPM learning algorithm obtains good estimates of the 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 loglikelihood 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 this provides more information for localizing the timing of process instances.

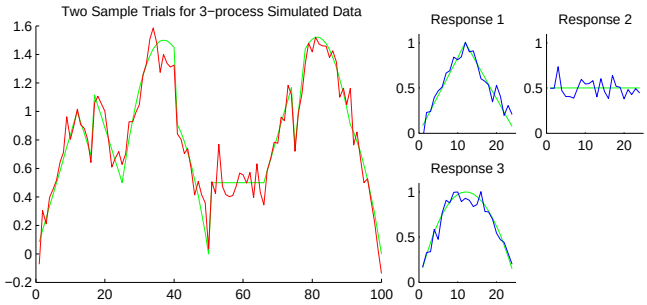

Figure 2: Learned versus true process responses: synthetic data. Plots on the right show learned response signatures (blue lines) for three processes superimposed on the true response signatures (green lines). This HPM was learned from the synthesized data shown on the left, in red; the green line indicates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

3.2 Experiments with fMRI data

In this fMRI study [6], human subjects were presented a sequence of 40 trials. In half the trials they were presented a picture for 4 sec, a blank screen for 4 sec, then a sentence. Then they pressed a button to indicate whether the sentence correctly described the picture. In the remaining trials the sentence was presented before the picture. Throughout, fMRI images of brain activity were captured every 500 msec.

We used three different HPMs to analyze this data. The first was a 2-process HPM which assumes the fMRI data is generated by a ReadSentence process and a ViewPicture process, each of which is instantiated immediately whenever the corresponding sentence or picture stimulus is presented, with a duration of 11 seconds. This is a typical duration for the fMRI response to neural activity (note this means the fMRI responses to the first and second stimuli overlap). We also considered a 3-process HPM which included the same ReadSentence and ViewPicture processes, plus a third Decide process (to model the subject's cognitive process of comparing the stimuli). The timing for ReadSentence and ViewPicture in this 3-process model were identical to the 2-process HPM, but the timing of the third Decide process was unspecified, with uniform priors on start times in an interval following the second stimulus. Finally we considered a model identical to the above 2-process HPM, but with process durations of 8 sec to assure the response signatures of processes did not overlap. We refer to this HPM model as the GNB model, because the non-overlapping responses make it equivalent to a Gaussian Naive Bayes classifier.

We trained each HPM and evaluated them using a leave-one-trial-out cross validation method. We measured their data loglikelihood and their classification accuracy when labeling each process as either ReadSentence or ViewPicture on the held-out data. The results are given in Table 1, for five human subjects. First note that both HPMs outperform the Gaussian Naive Bayes (GNB) model, in both data loglikelihood and classification accuracy. We take this as a promising sign of the superiority of HPMs over earlier classifier methods (e.g., [7]) for modeling cognitive processes.

Second, notice the 3-process HPM outperforms the 2-process HPM. 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 loglikelihood means that it is able to find useful structure in the data by incorporating the additional process.

We also applied HPMs to data from a second fMRI study in which subjects were presented a sequence of 120 words, one every 3-4 seconds, and decided whether the word was a noun or verb. We trained a two-process HPM, with processes ReadNoun and ReadVerb, each with duration 15 sec. This implies there are overlapping contributions from up to 5 distinct process instances 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 greatly overlapped fMRI 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.

4 Conclusion

We have presented HPMs to model hidden and temporally overlapping processes, along with algorithms for inference and learning. We have shown the robustness of HPMs with synthetic data experiments, and our results on real fMRI data show potential for HPMs as a new way to examine cognitive processes.

Our future work will improve our model in several ways. We will extend the model to handle parametric response forms, like the parametric hemodynamic response in [5]. We will allow real-valued offset times. Our model currently assumes white noise, but we plan to consider more general noise models. We will also explore approximate inference techniques to scale up HPMs. Additionally, we would like to allow variable-duration processes, timing dependencies between process instances, and domain-specific process parameters (e.g. whether a sentence was

Table 1: fMRI study: leave-one-trial-out cross validation results for GNB and HPM on the five subjects (A through E) exhibiting the highest accuracies and data log-likelihoods 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
accuracy GNB	0.725	0.750	0.725	0.637	0.750
accuracy 2-process HPM	0.750	0.875	0.700	0.675	0.787
accuracy 3-process HPM	0.775	0.875	0.738	0.637	0.812
loglikelihood GNB	-896.23541	-786.75823	-941.54912	-783.50593	-476.53631
loglikelihood 2-process HPM	-876.44947	-751.3732	-912.31519	-768.7222	-466.71741
loglikelihood 3-process HPM	-864.70878	-713.63435	-898.53191	-753.82864	-447.55965

affirmative or negative). Finally, we believe that HPMs solve a problem that is not specific to fMRI, and we are seeking additional appropriate domains.

Acknowledgments

We thank our colleagues Francisco Pereira and Marcel Just for helpful discussions throughout this work, and NSF, DARPA, and the Keck Foundation for their grant support.

References

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

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

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

[4] P. Højen-Sørensen, L. K. Hansen, and C. E. Rasmussen. Bayesian modelling of fMRI time series. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Proc. Conf. Advances in Neural Information Processing Systems, NIPS*, volume 12, pages 754–760, 2000.

[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] 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.

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