# Using Neural Imaging and Cognitive Modeling to Infer Mental States while Using an Intelligent Tutoring System

Jon M. Fincham, John R. Anderson, Shawn Betts and Jennifer L. Ferris
{fincham, ja+}@cmu.edu
Psychology Department, Carnegie Mellon University

Abstract. Functional magnetic resonance imaging (fMRI) data were collected while students worked with a tutoring system that taught an algebra isomorph. A cognitive model predicted the distribution of solution times from measures of problem complexity. Separately, a linear discriminant analysis used fMRI data to predict whether or not students were engaged in problem solving. A hidden Markov algorithm merged these two sources of information to predict the mental states of students during problem-solving episodes. The algorithm was trained on data from one day of interaction and tested with data from a later day. In terms of predicting what state a student was in during any 2 second period, the algorithm achieved 87% accuracy on the training data and 83% accuracy on the test data. Further, the prediction accuracy using combined cognitive model and fMRI signal showed superadditivity of accuracies when using either cognitive model or fMRI signal alone.

## 1 Introduction

This paper reports an approach of integrating cognitive modeling and neural imaging data to facilitate student modeling in intelligent tutoring systems. Intelligent tutoring systems have proven to be effective in improving mathematical problem solving (14, 20). Their basic mode of operation is to track students while they solve problems and offer instruction based on this tracking. These tutors individualize instruction by two processes called model tracing and knowledge tracing. Model tracing uses a model of students' problem solving to interpret their actions. It tries to diagnose the student's intentions by finding a path of cognitive actions that match the observed behavior of the student. Given such a match, the tutoring system is able to provide real-time instruction individualized to where that student is in the problem. The second process, knowledge tracing, attempts to infer a student's level of mastery of targeted skills and selects new problems and instruction suited to that student's knowledge state. While the principle of individualizing instruction to a particular student holds great promise, the practice is limited by the ability to diagnose what the student is thinking. The only information available to a typical tutoring system comes from the actions that students take in the computer interface. Inferring cognitive state based on such potentially impoverished data can be difficult and brain imaging data might provide a useful augmentation. Recent research has reported a variety of successes in using brain imaging to identify what a person is thinking about (e.g., 7, 10,11,15) and identifying when mental states happen (e.g., 1, 12, 13).

While the methods described here could extend to knowledge tracing, this article will focus on model tracing where the goal is to identify the student's current mental state. Two features of the intelligent tutoring situation shaped our approach to the problem:

1. Given that instruction must be made available in real time, inferences about mental state can only use data up to the current point in time. While inferences of

mental state may become clearer after observing subsequent student behavior, these later data are unavailable for real-time prediction.

2. Model tracing algorithms are parameterized with pilot data and then used to predict the mental state of students in learning situations. Therefore, we trained our algorithm on one set of data and tested it on a later set.

While many distinctions can be made about mental states during the tutor interactions, we focused on two basic distinctions as a first assessment of the feasibility of the approach. The first distinction involved identifying periods of time when students were engaged in mathematical problem solving and periods of time when they were not. The second, more refined, distinction involved identifying what problem they were solving when they were engaged and, further, where they were in the solution of that problem. While one might think only the latter goal would be of instructional interest, detecting when students are engaged or disengaged during algebraic problem solving is by no means unimportant. A number of immediate applications exist for accurate diagnosis of student engagement. For instance, there are often long periods when students do not perform any action with the computer. It would be useful to know whether the student was engaged in the mathematical problem solving during such periods or was off task. If the student was engaged in algebraic problem solving despite lack of explicit progress the tutor might volunteer help. On the other hand, if the student was not engaged, the tutoring system might nudge the student to go back on task.

The research reported here used an experimental tutoring system described in Anderson (2) and Brunstein et al. (5) that teaches a complete curriculum for solving linear equations based on the classic algebra text of Foerster (8). The tutoring system has a

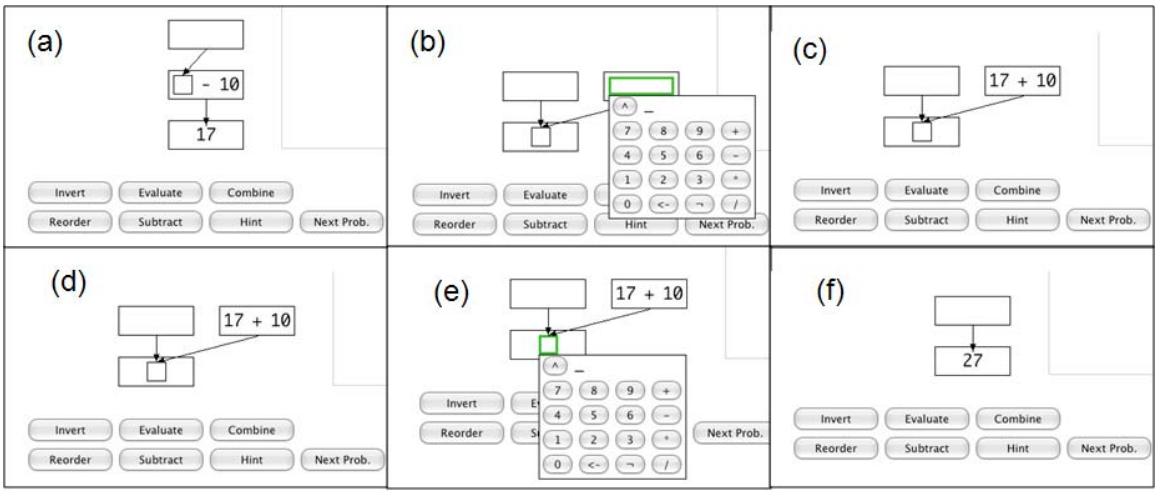


**Figure 1 Interface for equation solving isomorph. (a) The student starts out in a state with a data-flow equivalent of the equation x – 10 = 17. The student uses the mouse to select this equation and chooses the operation "Invert" from the menu. (b) A keypad comes up into which the student enters the result 17+10. (c) The transformation is complete. (d) The previous state (data-flow equivalent of x = 17+10) is repeated and the student selects 17+10 and chooses the operation "Evaluate". (e) A keypad comes up into which the student will type 27. (f) The evaluation is complete**.

minimalist design to facilitate experimental control and detailed data collection. It presents instruction, provides help when requested, and flags errors during problem solving. In addition to teaching linear equations to children, this system can be used to teach rules for transforming data-flow graphs that are isomorphic to linear equations. The data-flow system has been used to study learning with either children or adults and has the virtue of not interfering with instruction or knowledge of algebra. The experiment reported here uses this data-flow isomorph with an adult population. Figure 1 illustrates sequences of tutor interaction during a problem isomorphic to the simple linear equation $x - 10 = 17$. The interactions with the system are done with a mouse that selects parts of the problem to operate on, actions from a menu, and enters values from a displayed keypad.

## 2 Experiment

Twelve students went through a full curriculum based on the sections in the Foerster text for transforming and solving linear equations. The experiment spanned six days. On Day 0, students practiced evaluation and familiarized themselves with the interface. On Day 1, three critical sections were completed with functional magnetic resonance imaging (fMRI). On Days 2-4 more complex material was practiced outside of the fMRI scanner.

On Day 5 the three critical sections (with new problems) were repeated, again in the fMRI scanner. Each section on Days 1 and 5 involved 3 blocks during which they would solve 4 to 8 problems from the section. Some of the problems involved a single transformation-evaluation pair as in Figure 1 and others involved 2 pairs (problems studied on Days 2-4 could involve many more operations). Periods of enforced off-task time were created by inserting a 1-back task (17) after both transformation and evaluation steps. A total of 104 imaging blocks were collected on Day 1 and 106 were collected on Day 5 from the same 12 students. Average time for completion of a block was 207 2-second scans with a range from 110 to 349 scans. The duration was determined both by the number and difficulty of the problems in a block and by the students' speed.

Students solved 654 problems on Day 1 and 664 on Day 5. 76% of the problems on both days were solved with a perfect sequence of clicks. Most of the errors appeared to reflect interface slips and calculation errors rather than misconceptions. Each problem involved one or more of the following types of intervals:

1. Transformation (steps a-c in Figure 1): On Day 1 students averaged 8.2 scans with a standard deviation of 5.9 scans. On Day 5 the mean duration was 5.9 scans with a standard deviation of 4.1.
2. 1-back within a problem: This was controlled by the software and was always 6 scans.
3. Evaluation (steps d-f in Figure 1): Students took a mean of 4.9 scans on Day 1 with a standard deviation of 3.6; they took 3.8 scans on Day 5 with a standard deviation of 2.7.
4. Between Problem Transition: This involved 6 scans of 1-back, a variable interval determined by how long it took students to click a button saying they were done, and 2 scans of a fixation cross before the next problem. This averaged 9.1 scans with a standard deviation of 1.5 scans on both days.

In addition there were 2 scans of a fixation cross before the first problem in a block and a number of scans at the end which included a final 1-back but also a highly variable period of 6 to 62 scans before the scanner stopped. The mean of this end period was 11.0 scans and the standard deviation was 6.5 scans.

The student-controlled intervals 1 and 3 show a considerable range, varying from a minimum of 1 scan to a maximum of 54 scans. Anderson (2) and Anderson et al. (3) describe a cognitive model that explains much of this variance. For the current purpose of showing how to integrate a cognitive model and fMRI data, the complexity of that model would distract from the basic points. Therefore, we instead adapt a keystroke model (6) based on the fact that cognitive complexity is often correlated with complexity in terms of physical actions. Such models can miss variability that is due to more complex factors, but counting physical actions is often a good predictor.

We will use number of mouse clicks as our measure of complexity. As an example of the range in mouse clicks – it takes 15 clicks in the tutor interface to accomplish the following transformation[1]:

$$\frac{1000 * X}{-10} \Rightarrow \frac{1000}{-10} * \frac{X}{-10}$$

but only 5 clicks to accomplish the evaluation:

$$X = 7 - 5 \Rightarrow X = 2$$

Transformation steps take longer than evaluation steps because they require more clicks (average 10.4 clicks versus 6.8). Figure 2 illustrates the systematic relationship that exists between mouse clicks required to accomplish an operation and the time that the operation took. The average scans per mouse click decreases from .77 scans on Day 1 to .57 on Day 5. On the other hand the average ratio shows little difference between transformations (.69 scans) and evaluations (.65 scans) and so Figure 2 is averaged over transformations and evaluations. As the figure illustrates, the number of scans for a given number of mouse clicks is approximately distributed as a log-normal distribution. Log-normal distributions estimated from Day 1 were part of the algorithm for identifying mental state. The only adjustment for Day 5 was to speed up the mean of the distribution by a constant 0.7 factor (based on Anderson (2), model in that volume figure 5.7) to reflect learning. Thus, the prediction for Day 5 is .77*.7 = .54 scans per click.

### 2.1 Imaging Data

Anderson et al. (3) describe an effort to relate fMRI activity in predefined brain regions to a cognitive model for this task. However, as with the latency data, the approach here makes minimal theoretical assumptions. We defined 408 regions of interest (ROIs), each approximately a cube with sides of 1.3 cm that cover the entire brain. For each scan for each region, we calculated the percent change in the fMRI signal for that scan from a baseline defined as the average magnitude of all the preceding scans in that block. We used this signal to identify On periods when a student was engaged in problem solving (evaluation and transformation in Figure 1) versus Off periods when the student was engaged in n-back or other beginning and ending activities. A linear discriminant analysis

[1] For brevity we will give the standard algebraic equivalent of data-flow graphs.

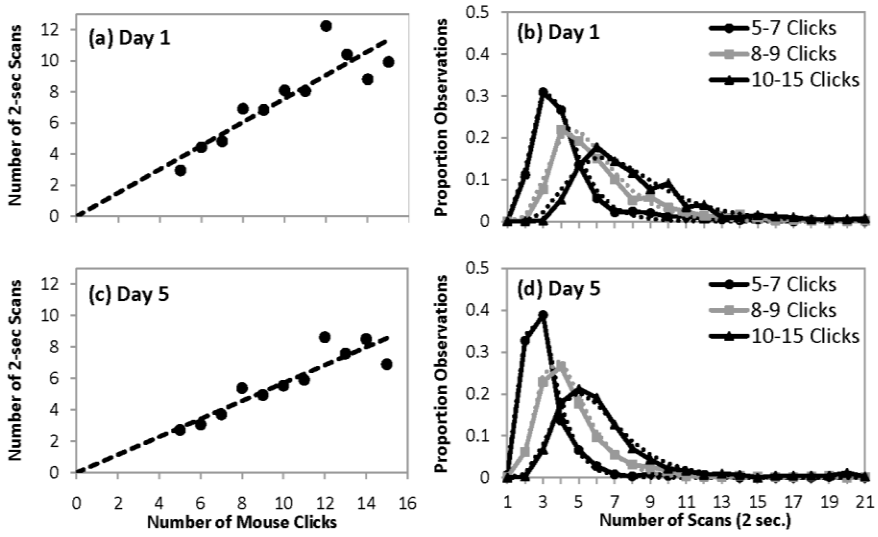


**Figure 2. (a) and (c): The relationship between number of clicks and duration of problem solving in terms of number of 2-sec scans. (b) and (d): Distributions of number of scans for different numbers of clicks and log-normal distributions fitted to these.**

was trained on the group data from Day 1 to classify the pattern of activity in the 408 regions as reflecting an On scan or an Off scan.Figure 3a shows how accuracy of classifying a target scan varied with the distance between the target scan and the scan whose activity was used to predict it. It plots a d-prime measure (9), which is calculated from the z-transforms of hit and false alarm rates. So, for instance, using the activity 2 scans after the target scan, 91% of the 7761 Day 5 On scans were correctly categorized and 16% of 11835 Off scans were false alarmed yielding a d-prime of 2.34. Figure 3 shows that best prediction is obtained using activity 2 scans or 4 seconds after the target scan. Such a lag is to be expected given the 4-5 second delay in the hemodynamic response. The d-prime measure never goes down to zero reflecting the residual statistical structure in the data.

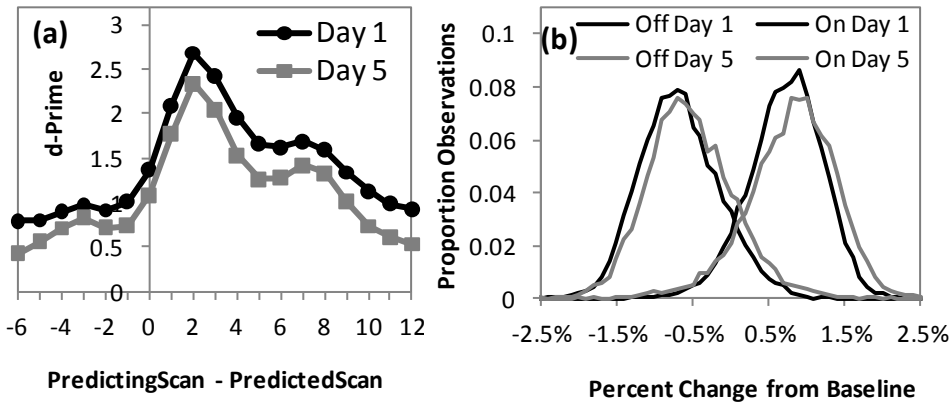


**Figure 3. (a) Accuracy of classification as a function of the offset between the scan whose activity is being used and the scan whose state is being predicted. (b) Distribution of fMRI signal changes for Day 1 and Day 5 On and Off scans using an offset of 2. All 408 regions are used.**

While we will report on the results using a lag of 0, the main application will use the optimal lag 2 results – meaning it was 4 seconds behind the student. Little loss occurs in d-prime going from training data to predicted data. The relatively large number of scans (21,826 on Day 1 and 19,596 on Day 5) avoids overfitting with even 408 regions. While our goal is to go from Day 1 to Day 5, the results are almost identical if we use Day 5 for training and Day 1 for testing. The weights estimated for the 408 regions can be normalized (to have a sum of squares of 1) and used to extract an aggregate signal from the brain. This is shown in Figure 3b for the On and Off scans on the two days.

**2.2 Predicting Student State**.

Predicting whether a student is engaged in problem solving is a long way from predicting what the student is actually thinking. As a first step to this we took up the challenge of determining which problem a student was working on in a block and where a student was in the problem. This amounts to predicting what equation the student is looking at. Figure 4.1 illustrates an example from a student working on a set of 5 equations. As the figure illustrates, each equation goes through 4 forms on the way to the solution: the first and third require transformation operations while the second and fourth require evaluation operations (see Figure 1). Adding in the 21 Off states between forms there are 41 states. Consider the task of predicting the student state on scan 200. Information available to the algorithm includes the 5 problems, the distributions of lengths for the various states, and that there are 41 states in all. The classifier additionally provides the probability that each of scans 1-200 came from an On state or an Off state. The algorithm must integrate this knowledge into a prediction about what state, from 1 to 41, the student is in at scan 200.

A key concept is an interpretation. An interpretation assigns the m scans to some sequence of the states 1, 2, …, r with the constraint that this is a monotonic non-decreasing sequence beginning with 1. For example, assigning 10 scans each to the states 1 to 20 would be one interpretation of the first 200 scans in Figure 4.1. Using the naïve Bayes rule, the probability of any such interpretation, *I*, can be calculated as the product of prior probability determined by the interval lengths and the conditional probabilities of the fMRI signals given the assignment of scans to On and Off states:

$$p(I \mid fMRI) \propto \left[ S_r(a_r) \prod_{k=1}^{r-1} p_k(a_k) \right] * \left[ \prod_{j=3}^{m+2} p(fMRI_j \mid I) \right]$$

The first term in the product is the prior probability and the product in the second term is the conditional probability. The terms pk(ak) in the prior probability are the probabilities that the kth interval is of length k and Sr(ar) is the probability the rth interval surviving at least as long as ar. These can be determined from Figure 2 for On intervals and from the experimental software for Off intervals. The second term contains p(fMRIj|I), which are the probabilities for the combined fMRI signal on scan j+2 given *I*'s assignment of scan j to an On or a Off state. The linear classifier determines these from normal distributions fitted to the curves in Figure 3b.

To calculate the probability that a student is in state r on any scan m one needs to sum the probabilities of all interpretations of length m that end in state r. This can be efficiently calculated by a variation of the forward algorithm associated with hidden Markov models[2] (HMMs, 19) . The predicted state is the highest probability state. The most common HMM algorithm is the Viterbi algorithm, a dynamic programming algorithm that requires knowing the end of the event sequence to constrain interpretations of the events. The algorithm we use is an extension of the forward algorithm associated with HMMs and does not require knowledge of the end of the event sequence. As such it can be used in real time and is simpler. Figure 4.1 illustrates the performance of this algorithm on a block of problems solved by the first student. Figure 4.1a shows the 20 forms of the 5 equations. Starting in an Off state, going through 20 On states, and ending in an Off state, the student goes through 41 states. Figure 4.1b illustrates in maroon the scans on which the algorithm predicts that the student is engaged on a particular equation form. Predictions are incorrect on 19 of the 241 scans but never off by more than 1 state. In 18 of these cases it is one scan late in predicting the state change and in 1 case it is one scan too early.

Going beyond showing 1 student during 1 block, Figure 4.2 shows the average performance over the 104 blocks on Day 1 and the 106 blocks on Day 5.

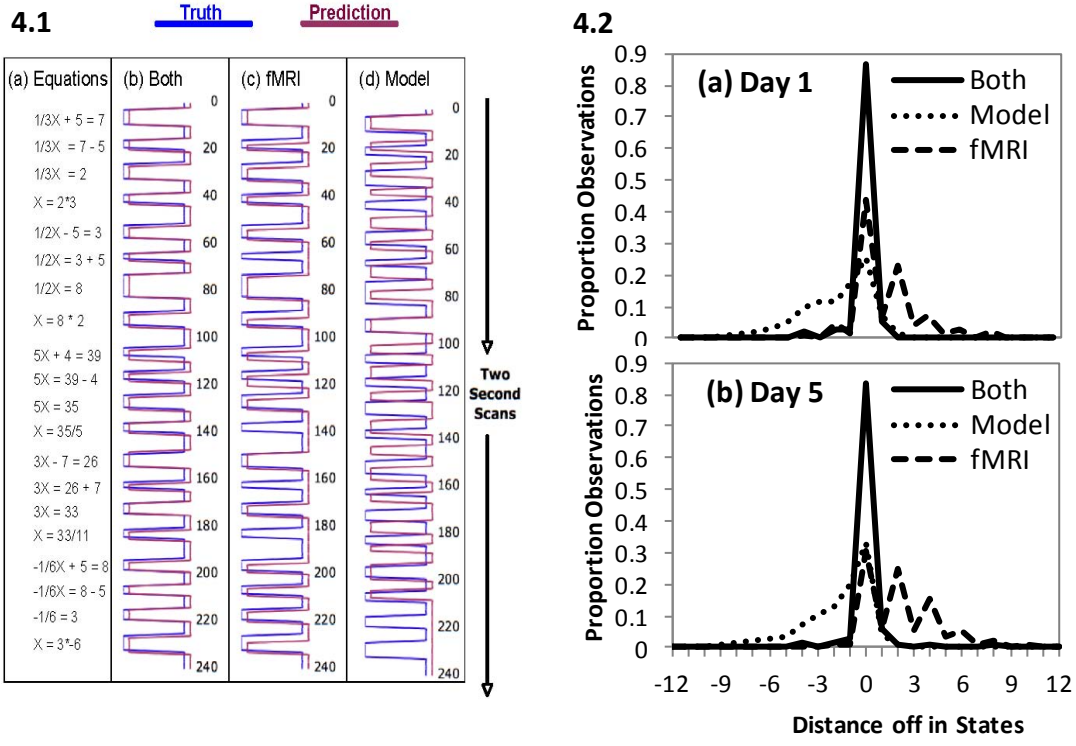


**Figure 4. (4.1) An example of an experimental block and its interpretations. The sequence of equations is shown in column a. Columns b, c, and d compares attempts at predicting the states with both fMRI and model, just fMRI, or just model. On scans (when an equation is on the screen) are to the left and Off times (when no equation is on the screen) are to the right. (4.2) Performance, measured as the distance between the actual state and the predicted state, using both cognitive model and fMRI, just fMRI, or just a cognitive model on (a) Day and (b) Day 5.**

[2] Since the states are not directly observable and their durations are variable our model is technically a hidden semi-Markov process (16).

Performance is measured in terms of the distance between the actual and predicted states in the linear sequence of states in a block. A difference of 0 indicates that the algorithm correctly predicted the state of the scan, negative values are predicting the state too early, and positive values are predicting the state too late. The performance of the algorithm is given in the curve labeled "Both". On Day 1 it correctly identifies 86.6% of the 22138 scans and is within 1 state (usually meaning the same problem) on 94.4% of the scans. Since all parameters are estimated on Day 1, the performance on Day 5 represents true prediction: It correctly identifies 83.4% of the 19914 scans on Day 5 and is within 1 state on 92.5% of the scans. To provide some comparisons, Figure 4.2 shows how well the algorithm could do given only the simple behavioral model or only the fMRI signal.

The fMRI-only algorithm ignores the information relating mouse clicks to duration and sets the probability of all lengths of intervals to be equal. In this case, the algorithm tends to keep assigning scans to the current state until a signal comes in that is more probable from the other state. This algorithm gets 43.9% of the Day 1 scans and 30.6% of the Day 5 scans. It is within 1 scan on 51.8% of the Day 1 scans and 37.3% of the Day 5 scans. Figure 4.1c illustrates typical behavior -- it tends to miss pairs of states. This leads to the jagged functions in Figure 4.2 with rises for each even offset above 0.

The model-only algorithm ignored the fMRI data and set the probability of all signals in all states to be equal. Figure 4.1d illustrates typical behavior. It starts out relatively in sync but becomes more and more off and erratic over time. It is correct on 21.9% of the Day 1 scans and 50.4% of the Day 2 scans. It is within 1 scan on 32.9% of the Day 1 scans and 56.9% of the Day 5 scans.

The performances of the fMRI-only and model-only methods are quite dismal. Successful performance requires knowledge of the probabilities of both different interval lengths and different fMRI signals.

## Conclusions

The current research attempted to hold true to two realities of tutor-based approaches to instruction. First, the model-tracing algorithm must be parameterized on the basis of pilot data and then be applied in a later situation. In the current work, the algorithm were parameterized with an early data set and tested on a later data set. Second, the model-tracing algorithm must provide actionable diagnosis in real time – it cannot wait until all the data are in before delivering its diagnosis. In our case, the algorithm provided diagnosis about the student's mental state in almost real time with a 4 second lag. Knowledge tracing, which uses diagnosis of current student problem solving to choose later problems, does not have to act in real time and can wait until the end of the problem sequence to diagnose student states during the sequence. In this case one could also use the Viterbi algorithm for HMMs (19) that takes advantage of the knowledge of the end of the sequence to achieve higher accuracy. On this data set the Viterbi algorithm is able to achieve 94.1% accuracy on Day 1 and 88.5% accuracy on Day 2. Morever, prediction accuracy using both information sources was substantially greater than using either data

source alone. A Bayesian analysis can explain the basis of the apparent superadditivity of prediction accuracy when using the combined information sources. The odds of a scan being On given the model and the fMRI signal can be expressed:

**Odds(On | Model & Signal) = Odds(On | Model)* Likelihood-ratio(Signal| On & Model)**

If

**(a) Likelihood-ratio(Signal| On & Model) = Likelihood-ratio(Signal| On) --** that is, the signal magnitude depends only on whether the state is On,

**(b) Odds(On) = 1** -- that is, that On scans and Off scans are equally frequent, which is approximately true, and therefore Likelihood-ratio(Signal| On) = Odds(On| Signal),

then the equation above can be rewritten

**Odds(On | Model & Signal) = Odds(On | Model)* Odds(On| Signal)**,

or by inverting the odds

**Odds(Off | Model & Signal) = Odds(Off| Model)* Odds(Off| Signal)**.

These two equations show there is a multiplicative relationship in the Odds(Correct Acceptance) and Odds(Correct Rejection). Increasing either the strength of the signal or the strength of the model multiplies the effectiveness of the other factor

This experiment has shown that it is possible to combine brain imaging data with a cognitive model to provide a fairly accurate diagnosis of where a student is in episodes that last as long as 10 minutes. Moreover, prediction accuracy using both information sources was substantially greater than using either source alone. The performance in Figure 4.2 is by no means the highest level of performance that could be achieved. Performance depends on how narrow the distributions of state durations are (Figures 2b and 2d) and the degree of separation between the signals from different states (Figure 3b). The model leading to the distributions of state durations was deliberately simple, being informed only by number of clicks and a general learning decrease of .7 from Day 1 to Day 5. More sophisticated student models like those in the cognitive tutors would allow us to track specific students and their difficulties leading to much tighter distributions of state durations. On the data side, improvement in brain imaging interpretation would lead to greater separation of signals. Finally, other data like eye movements could provide additional features for a multivariate pattern analysis.

## References


[1] Adelnour, F. & Huppert, T. (2009). Real-time Imaging of human brain function by near-infrared spectroscopy using an adaptive general linear model. NeuroImage, 46, 133-143.

[2] Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? New York: Oxford University Press.

[3] Anderson, J. R., Betts, S. A., Ferris, J. L., & Fincham, J. M. (in press). Can Neural Imaging be Used to Investigate Learning in an Educational Task? In J. Staszewski (Ed.) Expertise and Skill Acquisition: The Impact of William G. Chase.

[4] Anderson, J. R., Qin, Y., Sohn, M-H., Stenger, V. A. & Carter, C. S. (2003). An information-processing model of the BOLD response in symbol manipulation tasks. Psychonomic Bulletin & Review. 10, 241-261.

[5] Brunstein, A., Betts, S., & Anderson, J. R. (2009). Practice enables successful learning under minimal guidance. Journal of Educational Psychology, 101, 790-802.

[6] Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Erlbaum.

[7] Davatzikos, C., Ruparel, Fan, Y., Shen, D. G., Acharyya, M., Loughead, J. W., Gur, W. R., & Langleben, D. D. (2005). Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. Neuroimage, 28, 663–668.

[8] Foerster, P. A. (1990). Algebra I, 2nd Edition. Menlo Park, CA: Addison-Wesley Publishing.

[9] Green, D. M. & Swets, J. A. (1966). Signal Detection Theory and Psychophysics. New York, NY: Wiley.

[10] Haxby, J.V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science, 293, 2425–2430.

[11] Haynes, J. D. & Rees, G. (2005). Predicting the stream of consciousness from activity in human visual cortex. Current Trends in Biology, 15, 1301–1307.

[12] Haynes, J.D., Sakai, K., Rees, G. Gilbert, S., Frith, C., & Passingham, R. E. (2007). Reading hidden intentions in the human brain. Current Trends in Biology, 17, 323–328.

[13] Hutchinson, R., Mitchell, T.M., Niculescu, R.S., Keller, T. A., Rustandi, I. & Mitchell, T. M. (2009). Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using Hidden Process Models. NeuroImage, 46, 87-104.

[14] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 30-43.

[15] Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K-M., Malave, V. L., Mason, R. A., & Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. Science, 320, 1191-1195.

[16] Murphy. K. (2002). Hidden semi-Markov models. Technical Report, MIT AI Lab.

[17] Owen, A., Laird, A., & Bullmore, E. (2005). N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. Human Brain Mapping, 25, 46-59.

[18] Qin, Y., Anderson, J. R., Silk, E., Stenger, V. A., & Carter, C. S. (2004). The change of the brain activation patterns along with the children's practice in algebra equation solving. Proceedings of National Academy of Sciences,101, 5686-5691.

[19] Rabiner, R. E. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE, 77 (2): 257-286

[20] Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive Tutor: Applied research in mathematics education. Psychonomic Bulletin & Review, 14, 249-255.