Predicting Students' Retention of Facts from Feedback during Study

Robert Lindsey (robert.lindsey@colorado.edu)
Department of Computer Science, 430 UCB
University of Colorado, Boulder, CO 80309 USA

Owen Lewis (owen.lewis@colorado.edu)
Department of Applied Mathematics, 526 UCB
University of Colorado, Boulder, CO 80309 USA

Harold Pashler (hpashler@ucsd.edu)
Department of Psychology, 0109
University of California, San Diego, La Jolla, CA 92093 USA

Michael Mozer (mozer@colorado.edu)
Department of Computer Science, 430 UCB
University of Colorado, Boulder, CO 80309 USA

Abstract

Testing students as they study a set of facts is known to enhance their learning (Roediger & Karpicke, 2006). Testing also provides tutoring software with potentially valuable information regarding the extent to which a student has mastered study material. This information, consisting of recall accuracies and response latencies, can in principle be used by tutoring software to provide students with individualized instruction by allocating a student's time to the facts whose further study it predicts would provide greatest benefit. In this paper, we propose and evaluate several algorithms that tackle the benefit-prediction aspect of this goal. Each algorithm is tasked with calculating the likelihood a student will recall facts in the future given recall accuracy and response latencies observed in the past. The disparate algorithms we tried, which range from logistic regression to a Bayesian extension of the ACT-R declarative memory module, proved to all be roughly equivalent in their predictive power. Our modeling work demonstrates that, although response latency is predictive of future test performance, it yields no predictive power beyond that which is held in response accuracy.

Keywords: intelligent tutoring, ACT-R, Bayesian inference, fact learning

Introduction

An effective way to teach facts is to test students while they are studying (Roediger & Karpicke, 2006). For example, if a student is learning the meanings of foreign words, an appropriately designed tutoring system would display a foreign word, ask the student to guess the English translation, and then provide the correct answer. In this work, we consider the case where students undergo several rounds of this type of study. By convention, we refer to the group of rounds as a *study session*. At the end of a study session, students have had several encounters with each item being studied. In addition to promoting robust learning, testing students during study provides valuable information that can in principle be used to infer a student's current and future state of memory for the material. Through the use of a student's performance during study to predict recall at a subsequent test, informed decisions can be made about the degree to which individual facts would benefit from further study. In this paper, we explore algorithms to predict a student's future recall performance on specific facts using both the accuracy of the student's responses during study, and their response latencies—the time it took to produce the responses. In principle, other information is available as well, such as the nature of errors made and the student's willingness to guess a response. However, we restrict ourselves to accuracy and latency data because such data are independent of the domain and the study question format. Thus, we expect that algorithms that base their predictions on accuracy and latency data will be applicable to many domains.

Predicting future recall accuracy from observations during study can be posed as a machine learning problem. Given a group of students for whom we have made observations, we divide the students into "training" and "test" groups. The training group is used to build predictive models whose performance is later evaluated using the test group. We developed several predictive models and describe them later in this paper. Of particular interest is a method we call Bayesian ACT-R (BACT-R). It is based on the declarative memory module of the ACT-R cognitive architecture (Anderson, Byrne, Douglass, Lebiere, & Qin, 2004). The module has equations that interrelate response latency during study, accuracy during study, the time periods separating study sessions from one another and from the test, and the probability of a correct answer at test. However, these equations have a large number of free parameters which makes it challenging to use the model in a truly predictive manner. BACT-R is a method for using Bayesian techniques to infer a distribution over the free parameters, which makes it possible to use the ACT-R equations to predict future recall.

This paper is organized as follows: first, we describe the experiment from which we obtained accuracy and latency data for a group of students studying paired associates. Next, we describe BACT-R and three other models we built to predict student recall in the experiment. Finally, we evaluate and discuss the performance of the algorithms.

Data

Our data are from an unpublished experiment by Pashler, Mozer, and Wixted (unpublished) in which 56 undergraduates tried to learn the disciplines of 60 relatively obscure Nobel prize winners. In an initial pass through the material, subjects were shown the names of the prize-winners paired with their disciplines. Each winner-discipline pair was displayed for five seconds. For each prize winner's name, subjects were given either three or six study opportunities during which they could guess the discipline. For each guess, they received auditory feedback that signaled whether or not the guess was correct. If it was incorrect, the correct answer was displayed on the screen. For these study trials, subjects responded by pressing one of four keys on a keyboard (the experiment involved only three disciplines, and a fourth key indicated "no guess"). During study, both the accuracies and latencies of the subjects' responses were recorded. Two weeks following study, subjects were evaluated in a cumulative test over all the material. The cumulative test was given in the same format as the study trials.

Approaches to Predicting Recall Performance

In our machine learning approach to predicting student recall at test, we split subjects into training and test groups. For both the training and test groups, we gave our algorithms access to response accuracies and latencies obtained during the study session. Additionally, we gave the algorithm access to the response accuracies at the cumulative test session for only the training group. In this section, we describe four increasingly complex algorithms designed to learn from the training group in order to make predictions about the test group.

We use the information from the training subjects to build a model that we apply to the test subjects to predict the probability that they will answer correctly when tested. The model is then evaluated on the test subjects: for each subject s in the test group and item i being learned, we use the model to predict the probability that s correctly recalled i, and compare this prediction to the observed accuracy. In the future, we will refer to s and i as a "subject-item pair."

Because all subjects learned the same set of items, it is possible to use the performance of the training group on a particular item to inform the predicted performance of the test group on this item. We chose to avoid methods that do this because they are restricted to situations where data are available for a large number of subjects learning the same set of items. In principle, the methods we explore here might work even if individuals learned different items chosen from the same domain.

Percentage Classifier

This was the simplest method we examined: given a subject-item pair, the predicted probability of a correct answer at test is the simply the fraction of correct answers given during study. Unlike the other methods we describe in this section, the percentage classifier does not use data from the training subjects — the only information came from the subject's own responses during study.

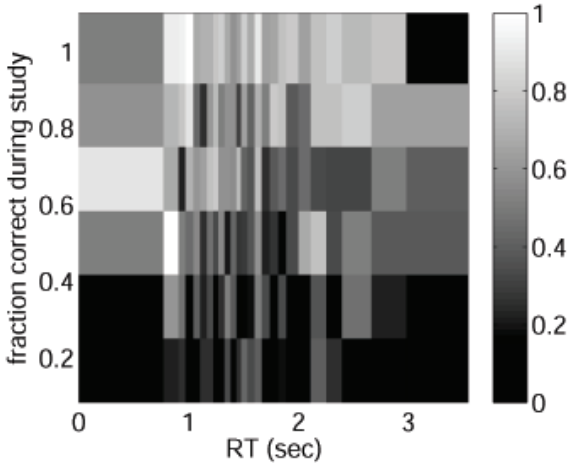

Figure 1: The grid used by the histogram classifier for subject-item pairs that had six study trials. Shading indicates the fraction of those subject-item pairs in the cell that had a correct answer at test. In this figure, the number of bins has been fixed. In practice, it is chosen by cross-validation and is unique to each test subject.

Histogram Classifier

For this method, we specified each subject-item pair by two numbers: the fraction of correct answers during study and the mean latency of the correct answers. We then formed two grids, one for the subject-item pairs that had three trials and another for the pairs that had six. The grids were formed in the following way: one axis had n numbers, such that each interval between two successive numbers contained an equal number of the mean latencies for the training set. n is a parameter of the model and was chosen by cross-validation. The other axis contained either four (for the three-session grid) or seven (for the six-session grid) numbers, such that each interval between two successive numbers contained exactly one of the possible fractions of correct answers. Each training example could then be placed in exactly one of the grid cells. For each cell, we found the number of training examples that fell within the cell and how many of these corresponded to a correct answer at evaluation. This enabled us to find, for each cell, a fraction correct. Given a test subject-item pair, we then found which cell it would fall into based on study performance and predicted that its probability of being correct at evaluation would be that cell's fraction correct. Figure 1 shows the grid for the six-trial case. Note that to display the figure, we had to fix the number of bins. In reality, since this number was chosen by cross-validation, it would be different for each test subject. In the grid shown in the figure, if a subject had a mean RT of 0.5 seconds for their correct answers and answered all study questions correctly, they would fall in the upper left hand cell, and have a predicted probability of future accuracy of about 0.6.

Logistic Regression

Logistic regression is a powerful prediction technique used in statistics and machine learning. In its simplest form, logistic regression takes the values of some number of predictor variables x_i (which may be either binary or continuous) corresponding to an input and then outputs a prediction of the probability that the input belongs to one of two classes. This probability of membership in one of the classes is given by:

$$f(x_1,\ldots,x_n) = \left[1+\exp(-\beta_0 - \sum_{i=1}^{n}\beta_i x_i)\right]^{-1}$$

The weights β_i are to be learned. β_0 is an offset term.

In this application, the predictor variables x_i are the latencies and accuracies obtained during study. More specifically, to predict the probability of a correct response at test for a subject-item pair with three study trials, we use six predictor variables. Three of these are binary and indicate whether each of the three answers given during study were correct or incorrect. The other three variables are the response latencies for the study answers and are therefore continuous. The predictor variables are constructed analogously for the six trial cases. The two classes are "correct answer at test" and "incorrect answer at test."

BACT-R

ACT-R is an influential cognitive architecture whose declarative memory module is often used to model recall follow a series of study sessions (e.g., Pavlik and Anderson (2008)). ACT-R assumes a separate trace is laid down each time an item is studied. Each trace decays according to a power law, t^{-D}, where t is the age of the memory and D is the decay rate. Following N study episodes, the activation for an item combines the trace strength of individual study episodes. It is governed by the equation:

$$A(\mathbf{t},D,B,c) = \log\left(\sum_{j=1}^{N} t_j^{-D}\right) + B + \varepsilon, \qquad \varepsilon \sim f(x;c)$$

where A is activation, B is a base activation level, ε is a noise term drawn from a logistic distribution with mean zero. That is, ε has the density function $f(x;c) = \frac{1}{4c}\operatorname{sech}^2\frac{x}{2c}$, where c is a free parameter. Recall probability is related to activation by:

$$\mathrm{P}(\text{correct recall}\,|\,A;\tau,c) = \left[1+\exp\left(\frac{\tau-A}{c}\right)\right]^{-1}$$

where τ is a free parameter. According to the model, latency (RT) is related to activation by:

$$\mathrm{RT}(A,F,f) = Fe^{-fA}$$

where F and f are free parameters. In total, there are six free parameters whose values we must estimate from the data: D,B,c,τ,F,f. Of these, we assume that c,τ,F,f are to be chosen for each subject-item pair, while the trace decay D and base-level activation term B are fixed for each subject.

For each subject-item pair we have a set of study-trial accuracies and latencies, and we can compute the likelihood of these data for any parameter vector. To do this, we plug the parameters into the equations to generate predictions for study trials and then compare these predictions to actual results of the study trials. More explicitly, we do likelihood-weighted sampling. For a given test subject, we take n_S samples from prior distributions of the six parameters. For each item, we compute the likelihood L of each set of parameters that have been generated. The final prediction of the probability of a correct answer at test is then:

$$\hat{P} = \sum_{i=1}^{n_S} P([D,B,c,\tau,F,f]_i)\ \frac{L([D,B,c,\tau,F,f]_i)}{\sum_{j=1}^{n_S} L([D,B,c,\tau,F,f]_j)}$$

where $\hat{P}$ is the prediction. The likelihood of a set of parameters with respect to a given subject-item pair is given by the product of its likelihood on each study trial:

$$\mathrm{L}(D,B,c,\tau,F,f) = \prod_{i=1}^{n_{trials}} l^i_{acc} l^i_{RT},$$

where i runs over study trials, and l_{acc} and l_{RT} denote the contribution to the likelihood of the accuracy and response latency. The l_{acc}

$$l^i_{acc} = \begin{cases} P(\text{correct recall}|\hat{A};\tau,c) & \text{if response } i \text{ is accurate} \\ 1-P(\text{correct recall}|\hat{A};\tau,c) & \text{otherwise} \end{cases}$$

Here, $\hat{A} = A(\mathbf{t},D,B,c)$.

$$l^i_{RT} = \begin{cases} \frac{1}{4c}\operatorname{sech}^2\frac{\hat{\varepsilon}}{2c} & \text{if response } i \text{ is accurate} \\ 1 & \text{otherwise} \end{cases}$$

where $\hat{\varepsilon} = \log\left(\frac{\mathrm{RT}^i}{\mathrm{RT}(\hat{A},F,f)}\right)$ and RT^i is the observed latency on the ith study trial. The intuition is that for a given set of parameters, we calculate how much noise would be necessary for these parameters to produce the observed latency and then take the likelihood to be the probability of observing this noise level. We used 250 samples for likelihood-weighted sampling. We found that increasing this number did not noticeably improve performance. One implementation detail should be noted: since the interval between study and test is so much larger than the interval between study sessions, we followed Pavlik and Anderson (2008) and compressed the interval between study and test into what they call "psychological time" via a small multiplicative factor.

To define priors for the six parameters, we use the fact that the framework above allows us to find, for each subject, maximum likelihood estimates for the parameter values. We do this for a group of training subjects and compile the results in a histogram. We then fit the results for each parameter to a probability distribution which is then that parameter's prior. In practice, the optimization routine we used to do the likelihood maximization did not converge for all subjects. The

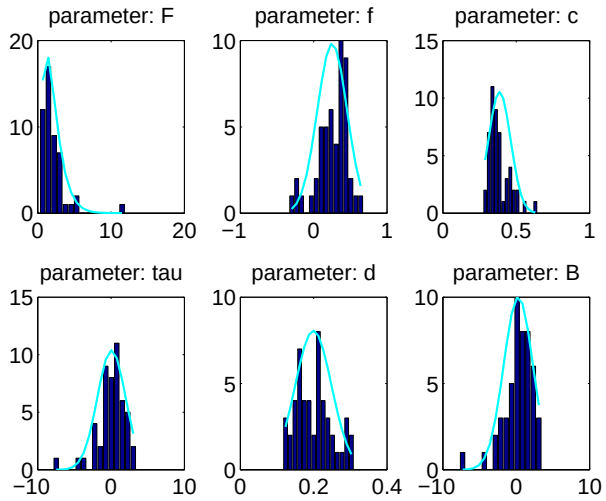

Figure 2: A set priors used in BACT-R. To set these priors, we find the maximum likelihood parameter values for each of the subjects in the training group, compile these estimates into histograms, and then fit the data for each parameter to a continuous probability distribution.

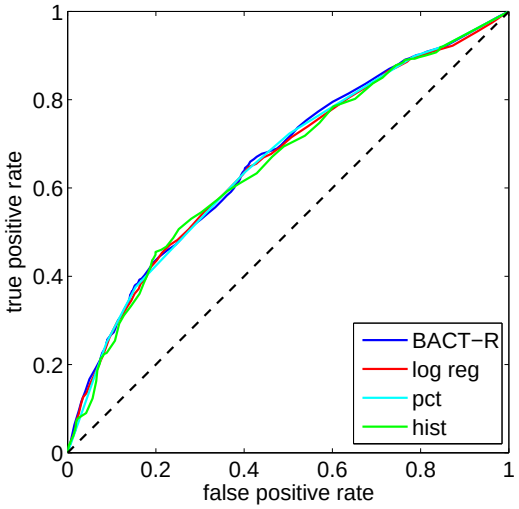

Figure 3: ROC curves for the methods we tried. A comparison shows that all methods perform similarly.

subjects for which it failed to converge were left out of the calculation of the prior. A set of priors, together with the histograms used to define them, are shown in Figure 2. This figure shows that the histograms were generally sharply peaked.

Results

To evaluate the different methods we tried, we used leave-one-out cross-validation. Each subject in turn was held out as a test subject and a prediction for that subject was made by models trained on all the other subjects. This prediction takes the form of a probability between zero and one. Because the data with which we have to compare these predictions are binary — a subject's response is either correct or incorrect — we thresholded the probability so that the predictions also become binary. After thresholding, the models' predictions are either true positive, false positive, true negative, or false negative. Adjusting the threshold changes the number of predictions that fall into each of these categories. In Figures 3-8 (to be described shortly), we summarize the threshold manipulation with an ROC curve, which plots the false positive rate versus the true positive rate for various thresholds. If the ROC curve falls exactly on the dashed diagonal line in the figures, then the method achieves results equivalent to chance prediction. In general, the more bowed the ROC curve, the better the performance of the model.

Comparison of methods

The results obtained by the various methods we tried are shown in Figure 3. As this figure shows, all the methods performed almost equally well. In particular, BACT-R did not outperform other methods we tried. It is interesting to note that this implies that the *order* of correct and incorrect responses, which is information to which BACT-R had access and the percentage classifier did not, seems not to have enabled BACT-R to outperform the percentage classifier. Of course, this does not necessarily mean that there is no useful information contained in the order data.

Relative Importance of Latency and Accuracy Information

We next examined how much information, if any, is contained in the latency data. Our findings are mixed. On the one hand, logistic regression and BACT-R performed just as well with the latency information removed as with it included (see Figures 4 and 5, respectively). On the other hand, when provided only with latency information, logistic regression yielded results significantly better than chance (Figure 4).

We also examined the weights given by logistic regression to latency and accuracy features. The inputs to logistic regression are normalized so that it is meaningful to compare the magnitudes of these weights. The mean magnitudes of the weights for accuracy and latency data are 0.3884 and 0.0751, respectively. The mean weight for the latencies is considerably smaller than the mean weight for the accuracies; it is not negligible. Thus, there is information in the latencies, but it is to a large extent redundant with the information from the accuracies.

The fact that latency information does not improve the performance of our methods may shed some light on our methods performing equivalently: no method took advantage of the latency information; all the information present in the accuracy information reduced to the percentage correct during study. Therefore, all methods did almost exactly as well as the percentage classifier.

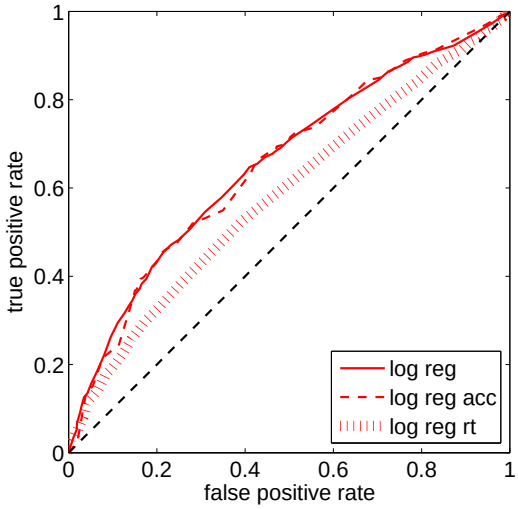

Figure 4: ROC curves for logistic regression, when the model was trained with all available data ("log reg"), only accuracy data ("log reg acc"), and only latency data ("log reg RT"). Removing the latency information does not degrade logistic regression's performance. However, using only the latency information gives results that are significantly better than random. We conclude that the latencies contain information, but that this information is redundant with the accuracy information, and does not help with classification.

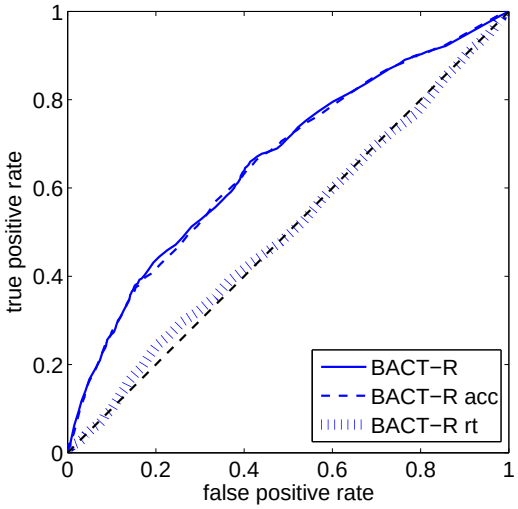

Figure 5: ROC curves for BACT-R when the method uses all available data, only the accuracy data, and only latency data. As with logistic regression (Figure 4), removing latencies does not noticeably hurt the performance of BACT-R. Using only latencies with BACT-R gives worse performance than it does with logistic regression.

Number of Study Trials

Figure 6 shows the performance of BACT-R when restricted to only the three- or the six-trial study conditions.

As expected, BACT-R performed better with six trials than with three, but the difference is not drastic. This is significant because it rules out the possibility that the BACT-R's performance was being dragged down by the three-session cases.

Another experiment we did involved applying logistic regression to only the first study session. In general, we have data from either three or six study trials for each subject-item pair. For this experiment, we used only the first of these. Apart from this, logistic regression was applied in the same way as before. The motivation for this experiment was the hypothesis that even if the accuracy information dominated the latency information when we used all the trials available, perhaps it would contribute more if we used only one trial. In fact, this was what we observed, as is shown in Figure 7, which indicates that, in the one-trial case, adding the latency information to the accuracy information gives a substantial improvement in performance. In addition, we see that it is possible to get reasonably good predictive performance even when we use information from only one trial.

Effect of Priors on BACT-R

In order to examine how much information was contained in the priors we used for BACT-R, we tried replacing the priors chosen by maximum likelihood with uniform priors having mean zero and length four, values that were chosen heuristically based on Figure 2. As Figure 8 shows, the results were noticeably worse than the results obtained with the maximum likelihood priors. This is a validation of the Bayesian approach, since it shows that the performance of the model was due, at least in part, to the knowledge contained in the prior distributions used for the parameters.

Variants

In addition to the methods described above, we tried several variants. For example, we tried replacing raw latencies with z-scores and including latencies from incorrect trials. We also tried assigning greater weight to information from later trials, since these were closer to the test time. No variant we tried significantly altered the performance of the models.

Discussion

Testing students as they study facts is known to be better than just having them reread the facts (Roediger & Karpicke, 2006). Testing has a side benefit: it produces feedback from the student which potentially could inform an intelligent tutoring system about how well the student has learned the facts. In this work, we described an experiment in which feedback was collected from students learning to identify the disciplines of 60 Nobel Prize winners. This feedback took the form of response accuracy and latency during a study session in which each fact was reviewed multiple times. Using data from the study session, we are able to predict memory for individual facts after a two-week retention interval.

We found that latency data alone was predictive. To the best of our knowledge, this finding has not been reported before in modeling literature. However, we also found that

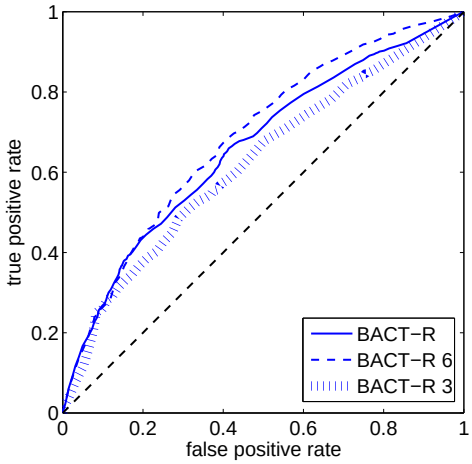

Figure 6: A comparison of the performance of BACT-R on the three-trial and six-trial subject-item pairs. Since six study trials give more feedback than three study trials, we expected BACT-R to perform better for these cases. As the figure shows, this is what we observed. Also as expected, we see that the three-study trial cases gave worse performance. However, BACT-R's performance on the three-study trial cases was not sufficiently degraded to conclude that these trials are responsible for BACT-R's inability to outperform the other methods we studied.

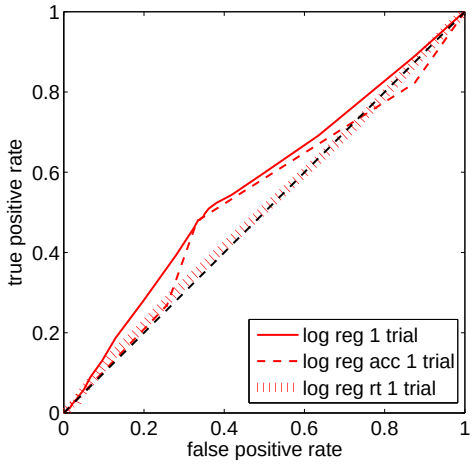

Figure 7: ROC curves for logistic regression when this method was applied to data from only the first study trial for each subject-item pair. If we look at only one study trial, we see that using latency information gives a substantial improvement in performance over the model trained with accuracy data alone. We also observe that, when using both pieces of data, we obtain reasonably good prediction performance, even on the basis of only one study trial.

adding latency data to accuracy data did not improve the performance of our models, suggesting that the latency information was redundant with the accuracy information.

We found that all the predictive models had similar performance, including a model based on ACT-R's declarative memory module, which is one of the best developed and evaluated high-level theories of human memory. Although BACT-R did not outperform other models, we believe that the addition of Bayesian uncertainty integration to the ACT-R framework is a promising idea that should be explored in other contexts. We also believe that the use of latency information for prediction of future recall warrants further study, especially when the feedback data are sparse (e.g., Figure 7, which shows the benefit of latencies when we have feedback from only one trial). Further, it would be interesting to see if the latency information from an experiment specially designed to elicit fast latencies would be more informative than the latencies from this experiment.

In one sense, our conclusions are not astonishing: accuracy of recall during study predicts accuracy of recall at a subsequent test. However, it is important that we have made this intuitively obvious relationship quantitative and that we have explored multiple computational approaches that can exploit the relationship to make concrete predictions of future recall performance.

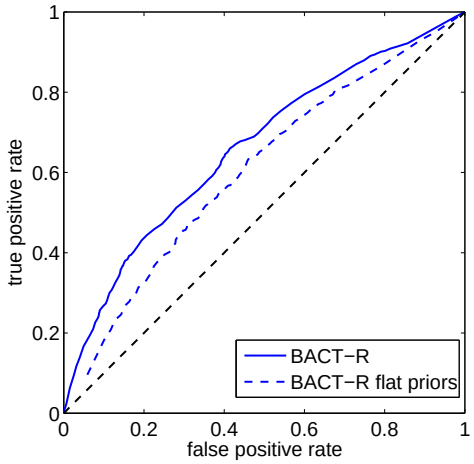

Figure 8: Using the maximum likelihood priors for BACT-R gives substantially better performance than using uniform priors.

References

Anderson, J. R., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-1050.

Pashler, H., Mozer, M., & Wixted, D. (unpublished). *Metrics of forgetting: weakening of associations versus skills.*

Pavlik, P. I., & Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. *J. Exp. Psych.: Applied*, *14*, 101–117.

Roediger, H., & Karpicke, J. (2006). The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science*, *1*, 181–210.