Predicting Improvement on Working Memory Tasks with Machine Learning Techniques: Part One

Jared Sylvester, Scott Weems & James Reggia
University of Maryland
Center for Advanced Study of Language and
Department of Computer Science

January 5, 2011

PURPOSE

To explore whether machine learning algorithms can be used to predict who will benefit the most from working memory training.

A recent TTO 3501 training study showed that peoples performance on a variety of working memory and language tasks can be improved by administration of a working memory task battery. This battery targets attentional and cognitive control mechanisms in the brain, thereby improving peoples ability to maintain and manipulate the information required for complex cognitive performance.

Our goal was to use machine learning methods to identify which, if any, individual characteristics of the trainees best predict who will show the most improvement. Prior to testing, all participants took an extensive demographic questionnaire. That information, along with pre-test scores prior to testing, was used as input to our models. Overall improvement as a result of the training was used as output, and several different learning algorithms were used to identify any relationships between these variables.

CONCLUSION

Overall, we were unable to identify any individual characteristics, or groups of characteristics, that accurately predict who will benefit the most from working memory training. However, it does appear that those who score lowest on pre-training working memory tasks are those who demonstrate the most overall improvement for those tasks.

Unfortunately, several factors made it impossible to identify any individual characteristics which predict working memory improvement following training. One was the small size of the dataset, while another was the widespread improvement of most subjects, including control subjects. Still, perhaps the most important factor was the weak correlation between most pre-test scores and later performance. In other words, there appeared to be little relationship between how well someone performed on the tasks prior to training, and how well they did on other tasks afterwards. The most notable exception was the observation that subjects who performed poorly on a particular task during pre-training, tended to show the most improvement for that particular task.

RELEVANCE

While our computational models did not find any characteristics which predicted who should benefit most from cognitive training, it is important to note that the approach remains potentially useful, and should be employed if future training provides larger data sets.

Although our lack of useable findings was disappointing, this research was still useful in that we now have an established approach for identifying factors predicting successful post-training performance. We have identified over ten machine learning methods for predicting working memory improvement based on demographic and pre-testing measures, any of which could be applied to future datasets. We have also developed useful methods for assessing performance of these machine learning techniques.

Predicting Improvement on Working Memory Tasks with Machine Learning Techniques: Part Two

Jared Sylvester, Scott Weems & James Reggia
University of Maryland
Center for Advanced Study of Language and
Department of Computer Science

January 5, 2011

Abstract

A recent experimental study has established that working memory training can improve a person's performance on cognitive tasks. This current report examines whether machine learning algorithms can be used to predict which subjects will improve. Multiple factors are used to make predictions, including both demographic data as well as performance on pre-training assessment tasks. Multiple machine learning algorithms of various types are assessed. In addition, two feature selection techniques are used to judge which factors are most useful in predicting improvement. The machine learning algorithms we examined were only marginally effective in predicting improvement. This was most likely due to the small size and lack of strong correlation between features and outcomes in the data set. Advanced algorithms rarely performed better than than simple guessing based on prior probabilities when considering the accuracy rate of predictions, though when judged on the basis of specificity and sensitivity they perform better. Ensemble algorithms did display some promise on this task, often out-performing other methods.

1. Motivation

A recent study at CASL demonstrated that some people improve their performance on a number of cognitive tasks after receiving working memory training when compared to control subject who do not receive the training [Atkins et al., Submitted; Bunting et al.,

2010]. What is as of yet unclear is why some subjects improve while others do not. Here we attempt to find a computational method for predicting a priori who will improve their performance on various cognitive tasks if they receive training. It could be quite valuable to predict who will benefit from training, since such training can be a costly undertaking, in both time and effort.

This study examines three separate questions. First, can we predict who will benefit from training? Second, which machine learning algorithms are best suited for this task? Finally, which characteristics of subjects are most useful in making this prediction?

2. Methods

The data for this study is drawn from the findings of TTO 3501. Specifically, we considered all subjects who participated in the working memory study training regime. Information about these subjects, including personal characteristics as well as their performance on all three batteries of testing (pre-training, post-training, and post-delay) were used to generate data sets for the various machine learning algorithms.

Input features of these data sets consisted of demographic data such as age, sex, and educational attainment, as well as the scores subjects received on pre-training assessments such as Listening Span and Cloze tasks. Subjects' performances were normalized according to their z-score. Multiple data sets were generated, each with the same input features, but differing in the class/output variable to be predicted.

Multiple tasks were used to generate different class variables to be predicted. In each case, the actual variable was binary: true if the subject improved their score after training on that task, and false if they did not. Additionally, both the post-training and post-delay scores were used to generate different data sets. Each task resulted in two data sets, one for whether there was pre-training to post-training improvement, the other for pre-training to post-delay improvement.

Furthermore, we used two definitions for what it meant to improve on a task. In the first phase of the study, the class variable was true if the subject's score went up at all between the former and latter administrations of the test. In the second phase, the subject's score had to change by at least as much as the average change among the control subjects. For example, the mean change in Rotation span scores between the first and second test administration for the untrained, control subjects was 1.67. A trained subject therefore had to improve their score by at least 1.67 for them to be put in the "improved" class; a decrease or an increase of less than 1.67 would be considered "unimproved."

Several machine learning methods were used to make predictions. The algorithms used were:

1. *Naïve Bayes*, a statistical classifier which assumes conditional independence among feature variables [John and Langley, 1995].

2. *Bayesian Network*, another statistical classifier that allows for conditional dependencies between variables [Pearl, 1986].

3. *Support Vector Machine*, a classifier which does a non-linear transformation of the feature space to try to achieve the maximum possible linear separability between classes [Platt, 1998].

4. *IBk*, an instance-based or nearest-neighbor classifier [Aha and Kibler, 1991]. (Our classifier used the $k = 5$ nearest neighbors.)

5. *C4.5*, a popular decision tree algorithm [Quinlan, 1993].

6. *Ripper*, an algorithm for induction of propositional logic rules [Cohen, 1995].

7. *Multilayer Perceptron*, an artificial neural network using error back propagation learning [Rumelhart et al., 1986].

8. *Bagging*, an ensemble or meta-classifier, which aggregates the predictions of many simple decision trees each using different features into a final prediction [Breiman, 1996].

9. *AdaBoost*, another decision tree ensemble, with each tree being trained on different subsets of the data set, weighted towards harder to classify instances [Freund and Schapire, 1996].

10. *Voting*, a method for combing predictions of multiple base classifiers, which can be trained with different algorithms [Kuncheva, 2004]. We used two methods of combining predictions: simple majority voting, and a vote weighted by the confidence of each sub-classifier.

11. *Stacking*, a method for combining predictions of several classifiers, possible of different types, by training a meta-classifier using the predictions of the underlying models as features [Wolpert, 1992]. We use both C4.5 and Naïve Bayes as the meta-classifier.

For both voting and stacking, the base classifiers consisted of one instance of each of the first nine methods listed above.

In addition to the above methods, we report the performance of the "ZeroR" classifier for comparison purposes only. ZeroR predicts that *all* instances will be in whichever class formed the majority of the training set.

Classifier success was judged by several different standards, principally the accuracy rate of predictions. In addition, sensitivity and specificity were assessed, because false positive and false negatives may have different impacts when predicting whether a person would benefit from training. Predicting a subject will benefit from training when they won't might be considered to be costlier, for instance, because that person will be needlessly put through a

lengthy training process. These values were calculated as:

$$specificity = \frac{\#\ true\ negatives}{\#\ true\ negatives\ +\ \#\ false\ positives}$$

$$sensitivity = \frac{\#\ true\ positives}{\#\ true\ positives\ +\ \#\ false\ negatives}$$

The other reason to consider sensitivity and specificity has to do with the class imbalances in the data sets. Many of the data sets were imbalanced, that is, they had significantly more samples in one class than the other. In such cases it becomes possible to produce high accuracy scores simply by always predicting the majority class for every test case. While accuracy may be high with such a strategy, sensitivity and specificity suffer.

All the results presented below were produced using ten fold cross validation, repeated ten times with different decile partitioning, for each data set.

To examine which features had the most predictive power, two measures were used. The first is the information gain ratio [Mantaras, 1991], a measure of how much a feature can reduce the entropy of a decision. The second is the CFS Subset algorithm [Hall, 1998], which attempts to select a subset of attributes that are highly correlated with the class variable while being uncorrelated with each other.

3. Results

3.1. Phase One

Table 1 shows the accuracy rates for each classifier when attempting to predict improvement on the post-training assessment. Table 2 shows the same classifiers accuracy when predicting post-delay improvement. Unsurprisingly, no classifier performed better on the post-delay data sets than the post-training ones.

Based on the findings in Bunting et al. [2010], we focused analysis on the Listening Span, O-Span and Symmetry Span data sets, since these are the tasks on which subjects showed the most benefit from training. The methods which showed the best accuracy on these data sets are Boosting, Bagging, SVM, and Voting. Of these, only the SVM is not an ensemble method. Note that while the accuracy of these classifiers is high the ZeroR classifier performs almost as well. This is possible because the data sets being used were small (44–45 data points) and imbalanced (class ratios were as high as 86%). This results in a data set with very few instances of one class on which to train and test, and as a result it is often nearly ignored by the learning algorithms.

ZeroR does not manage to match the other algorithms performance when considering the sensitivity and specificity of the classifiers (Tables 3–6). ZeroR always scores very well on one of these measures but very poorly on the other, depending on which class was more common in the training set. This is because always predicting the same outcome will be guaranteed, eliminating either false positives or false negatives, but doing so results in greatly increasing the other error type.

It is in specificity that classifiers can be seen differentiating themselves from a strategy of guessing. Starting with the methods already identified above as having high accuracy, the weighted average voting classifier attains the consistently highest specificities.

3.2. Phase Two

In attempts to equalize the class balance, we recreated these data sets, but required that the trained subjects' scores improve by at least the amount that the untrained, control subjects did. This resulted in much more balanced data on which to train, with the majority class being no higher than 62% of the data set.

Additionally, based on the findings of phase one, we eliminated the AFOQTR, Cloze and Verbal tasks, and added Rotation Span. We also dropped the Stacking methods from

consideration, as they are computationally expensive and showed only average performance on the previous phase.

Accuracy on both post-training and post-delay predictions can be seen in Tables 7 & 8. Interestingly, accuracy is now no better on the post-training data sets than the post-delay ones. (Note that while the accuracies are actually somewhat higher for the post-delay data sets, the accuracy of ZeroR is also higher compared to the post-training data sets, indicating that it is a somewhat "easier" prediction due to different class ratios.)

Rotation span appears particularly difficult to predict because it is significantly smaller than the other data sets. The other data sets had 44 or 45 data points, while rotation had only 36. More subjects had to be discarded because they did not have scores on the second and third administrations of the Rotation span task.

Of all five tasks, Decipher seems to be the one where the classifiers are most successful at out-performing ZeroR, followed by Operation Span. On Decipher, the algorithms which perform best are Ripper, Bagging, BayesNet, and Majority Voting. Bagging and voting are both ensemble methods that displayed good performance in the first phase as well.

Like the first phase results, the trained classifiers outperformed ZeroR's guessing strategy when judged by specificity (see Tables 11 & 12). While the first phase specificities were higher than ZeroR's, they were still low for many of the data sets. Classifiers recorded higher specificities in the second phase across the board.

Table 13 displays the four features with the highest information gain ratios for each of the ten data sets in the second phase of the study. For Operation Span and Decipher tasks, the single most informative features were the subject's score on the first administration of the task. These scores were negatively correlated with improvement on the tasks, for the most part. This means that the best way to predict who would improve would be to select all those with low initial scores, perhaps because they have the most "room for improvement."

Interestingly, a subject's first Cloze score appears to be a better predictor of their improvement on Listening Span than does their first Listening Span score. Similarly, Operation Span is a better predictor of Symmetry Span improvement than is the first Symmetry Span test. This suggests that if any of the pre-tests warrant further study for predictive ability, it would be Cloze and Operation Span.

It should also be noted that while these are the four features with the highest information gain ratios, these ratios are still extremely low, with even the highest ranked feature often having an information gain ratio under 0.1. This indicates that even these features offer only a weak ability to predict the class variable.

Table 14 shows the features selected by the CFS Subset algorithm. Note that this algorithm does not rank the features within the subset it selects, so the order in which the features are given is not an indication of their relative value, unlike in Table 13. The overall pattern is similar to that seen when using Information Gain Ratio. We see again that a subject's score on the first administration of a task is a useful predictor of their improvement, and that Operating Span and Cloze results are useful for predicting improvement on other tasks.

4. Discussion

The goal of this study was to investigate ways to predict a priori who would benefit from working memory training. The ability to preselect candidates who are more likely to show increased improvement from the training regime could potentially save a great deal of time and effort.

In order to identify good candidates for training, various machine learning algorithms were tested on data sets derived from demographic data about subjects as well as subjects' performance on a battery of pre-training cognitive tasks. These algorithms were of several basic types, including statistical, rule-inducing, instance-based, decision trees, and both homogenous and heterogenous ensembles. In addition, two algorithms for feature selection

were applied to judge which aspects of the subjects would be most useful for making predictions, independently of the actual learning algorithm that might be used.

We found that predicting trained subjects' improvement to be a very difficult task. One of the principle reasons for this is the small sample size of the data sets. This data was never intended to be used with machine learning methods as we have done here. Most machine learning algorithms are designed to work with at least several hundred training samples, not the approximately forty training samples given. The problem of small data sets was also exacerbated by the uneven class ratios in the first phase of the study, as the skewed data sets offered as few as six data points in the "unimproved" class on which to train. This made generalization very difficult.

Small data set sizes also made judgements about statistical significance problematic. A ten fold cross validation with 44 data points means there will be 40 training and four test instances. An algorithm with a mean accuracy of 75% could easily score two, three or four out of four on different test sets, leading to very large standard deviations in the accuracy, causing most results to seem statistically insignificant.

Another difficulty in making predictions from this data was that there were only very weak correlations between feature values and the class variables being predicted, if there was any correlation at all. Most features did not contribute to the predictive ability of the classifiers. In particular, most of the scores on the pre-training tasks did not correlate with subsequent improvements in other tasks, except for Operating Span and Cloze, as mentioned earlier. In addition there was a notable negative correlation between the pre-training score and later improvements on the same task, because the lower the initial score was the more opportunity a subject would have to increase it. Even this result is difficult to interpret, as it may be due to simple regression artifact. In other words, given a random fluctuation of scores (i.e., no true relationship between pre- and post-score measures), one would predict below average scores to increase and above average scores to decrease, due to normal regression to the

mean. In the end, it may therefore be difficult for any algorithm to discover a pattern in the data because no clear patterns may exist.

Because of the low predictive capacity of many of the variables considered, it would be prudent to begin any further investigations with a new data set, such as one drawn from the DLAB project, with an emphasis on feature selection. Many of the algorithms considered here, in particular the instance-based and statistical ones, would benefit from more rigorous culling of features.

It would also be useful, moving forward, to consider the varying impact of Type I and Type II errors from the outset when analyzing performance. For instance, considering the receiver operating characteristic curves of the different classifiers would give a move complete picture of the classifier performance than do scalars like the sensitivity and specificity alone. Some preliminary analysis of the area under the ROC curves suggests that the classifiers detailed above do considerable better than ZeroR, particularly on Decipher and Operation Span, even with the limitations of the data sets presented here.

References

D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

Sharona Atkins, Michael Doughtery, James Harbison, Jared Novick, Scott Weems, and Michael Bunting. Persistence and transferability of working memory training over time. Submitted.

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

Michael F. Bunting, Jared M. Novick, Michael R. Dougherty, J. Isaiah Harbison, Scott A. Weems, Sharona Atkins, Erika K. Hussey, Susan Teubner-Rhodes, Jeffrey Chrabaszcz, Alexei Smaliy, Carrie K. Clarady, and Ryan Corbett. Assessing the effects of cognitive training. Technical report, University of Maryland Center for Advanced Study of Languages, September 2010.

William W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

George H. John and Pat Langley. Estimating continuous distributions in bayesian clasifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, CA, 1995.

Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

R. Lopez Mantaras. A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6:81–92, 1991.

Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, 1998. MIT Press.

Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

David E. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In David E. Rumelhart and James L. McClelland, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*. MIT Press, Cambridge, MA, 1986.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

Table 1. Accuracy of predictions of post-training improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	53.0	55.1	43.1	57.4	62.8	61.0	62.3	54.9	61.1	60.2	59.2	52.0	61.8	62.2	57.9	62.8
Cloze	56.5	65.8	50.1	48.6	68.5	69.9	68.0	70.1	51.3	59.6	66.2	55.7	67.5	65.3	62.0	70.1
Decipher	74.5	67.1	73.9	75.6	69.5	70.7	64.3	68.2	75.1	74.4	65.4	66.5	72.0	73.0	70.4	75.6
Listening	64.0	65.1	56.8	53.9	65.2	57.8	54.8	50.7	62.5	54.6	45.4	57.1	61.1	62.0	57.4	65.2
O-Span	87.0	79.0	78.2	84.0	88.7	86.5	80.5	74.3	83.8	76.2	68.4	82.2	80.8	82.0	80.3	88.7
Symmetry	77.0	71.3	72.8	73.3	66.1	76.5	75.2	71.5	75.1	55.6	54.0	71.0	75.0	74.3	70.1	76.5
Verbal	54.0	65.5	69.8	61.1	77.7	76.4	71.2	72.5	84.3	84.9	81.4	75.9	75.6	78.8	75.0	84.9
mean	66.6	67.0	63.5	64.8	71.2	71.3	68.0	66.0	70.4	66.5	62.8	65.7	70.5	71.1		
max	87.0	79.0	78.2	84.0	88.7	86.5	80.5	74.3	84.3	84.9	81.4	82.2	80.8	82.0		

Table 2. Accuracy of predictions of post-delay improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	51.0	45.2	51.8	52.9	47.5	47.2	47.4	54.2	47.4	55.0	45.3	49.6	49.9	52.4	49.7	55.0
Cloze	52.0	58.9	69.0	62.3	63.0	69.6	71.8	72.3	61.9	67.6	64.0	62.6	68.5	67.4	66.1	72.3
Decipher	84.0	80.1	77.1	83.3	81.8	83.3	80.7	86.3	83.1	74.8	62.9	78.2	84.0	84.1	79.9	86.3
Listening	59.0	53.6	41.1	52.4	62.3	61.0	53.4	50.4	50.6	55.4	48.5	51.7	55.6	54.8	53.1	62.3
O-Span	79.5	68.2	75.6	77.5	70.2	77.6	72.0	68.9	75.4	70.6	52.4	72.1	76.9	76.6	71.8	77.6
Symmetry	79.5	73.9	69.9	77.3	74.1	74.5	68.6	72.2	78.9	69.2	67.6	73.8	74.2	74.4	72.9	78.9
Verbal	59.0	56.4	59.8	47.9	34.7	43.7	48.7	39.1	55.3	48.8	52.7	55.4	45.5	44.2	48.6	59.8
mean	66.3	62.3	63.5	64.8	61.9	65.3	63.2	63.3	64.6	63.0	56.2	63.3	64.9	64.8		
max	84.0	80.1	77.1	83.3	81.8	83.3	80.7	86.3	83.1	74.8	67.6	78.2	84.0	84.1		

Table 3. Sensitivity of predictions of post-training improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	0.00	0.50	0.48	0.47	0.63	0.61	0.64	0.48	0.57	0.56	0.59	0.47	0.55	0.64	0.55	0.64
Cloze	1.00	0.56	0.49	0.43	0.69	0.74	0.78	0.77	0.52	0.65	0.68	0.60	0.63	0.78	0.64	0.78
Decipher	1.00	0.77	0.78	0.88	0.84	0.95	0.83	0.77	0.88	0.82	0.73	0.82	0.82	0.95	0.83	0.95
Listening	1.00	0.78	0.74	0.76	0.79	0.82	0.75	0.60	0.84	0.65	0.54	0.72	0.73	0.84	0.74	0.84
O-Span	1.00	0.89	0.90	0.97	0.97	0.99	0.89	0.84	0.96	0.86	0.73	0.89	0.90	0.99	0.91	0.99
Symmetry	1.00	0.83	0.95	0.95	0.85	0.99	0.98	0.84	0.97	0.69	0.60	0.90	0.87	0.99	0.88	0.99
Verbal	1.00	0.67	0.72	0.74	0.85	0.78	0.76	0.72	0.90	0.88	0.83	0.82	0.79	0.90	0.80	0.90
mean	0.86	0.71	0.72	0.74	0.80	0.84	0.81	0.72	0.81	0.73	0.67	0.75	0.76	0.87		
max	1.00	0.89	0.95	0.97	0.97	0.99	0.98	0.84	0.97	0.88	0.83	0.90	0.90	0.99		

Table 4. Sensitivity of predictions of post-delay improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	1.00	0.43	0.44	0.50	0.48	0.45	0.55	0.55	0.48	0.58	0.51	0.53	0.50	0.58	0.50	0.58
Cloze	1.00	0.63	0.73	0.50	0.63	0.72	0.77	0.74	0.60	0.67	0.67	0.64	0.67	0.77	0.67	0.77
Decipher	1.00	0.93	0.92	0.99	0.93	0.99	0.95	0.94	0.99	0.81	0.71	0.92	0.92	0.99	0.92	0.99
Listening	1.00	0.57	0.58	0.70	0.77	0.80	0.70	0.62	0.63	0.63	0.57	0.62	0.66	0.80	0.66	0.80
O-Span	1.00	0.81	0.91	0.97	0.85	0.98	0.90	0.83	0.92	0.81	0.59	0.90	0.86	0.98	0.87	0.98
Symmetry	1.00	0.87	0.88	0.97	0.85	0.93	0.85	0.84	0.99	0.80	0.75	0.88	0.87	0.99	0.88	0.99
Verbal	1.00	0.67	0.65	0.70	0.50	0.64	0.77	0.55	0.71	0.57	0.57	0.70	0.63	0.77	0.65	0.77
mean	1.00	0.70	0.73	0.76	0.72	0.79	0.78	0.73	0.76	0.69	0.62	0.74	0.73	0.84		
max	1.00	0.93	0.92	0.99	0.93	0.99	0.95	0.94	0.99	0.81	0.75	0.92	0.92	0.99		

Table 5. Specificity of predictions of post-training improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	1.00	0.59	0.38	0.68	0.62	0.62	0.61	0.60	0.64	0.63	0.60	0.57	0.60	0.68	0.60	0.68
Cloze	0.00	0.77	0.50	0.56	0.67	0.63	0.54	0.62	0.50	0.52	0.63	0.50	0.59	0.77	0.60	0.77
Decipher	0.00	0.37	0.61	0.41	0.28	0.01	0.09	0.44	0.38	0.53	0.42	0.21	0.35	0.61	0.36	0.61
Listening	0.00	0.43	0.25	0.15	0.40	0.14	0.19	0.33	0.23	0.36	0.31	0.31	0.28	0.43	0.29	0.43
O-Span	0.00	0.17	0.02	0.00	0.32	0.00	0.22	0.13	0.00	0.17	0.27	0.35	0.13	0.32	0.16	0.35
Symmetry	0.00	0.32	0.00	0.00	0.02	0.00	0.00	0.30	0.00	0.10	0.32	0.09	0.11	0.32	0.12	0.32
Verbal	0.00	0.63	0.67	0.46	0.69	0.74	0.65	0.72	0.77	0.81	0.79	0.69	0.69	0.81	0.70	0.81
mean	0.14	0.47	0.35	0.32	0.43	0.31	0.33	0.45	0.36	0.45	0.48	0.39	0.39	0.56		
max	1.00	0.77	0.67	0.68	0.69	0.74	0.65	0.72	0.77	0.81	0.79	0.69	0.69	0.81		

Table 6. Specificity of predictions of post-delay improvement

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	StackNB	StackC45	VoteMaj	VoteAvg	*mean*	*max*
Afoqtr	0.00	0.47	0.60	0.56	0.47	0.50	0.39	0.53	0.47	0.51	0.40	0.46	0.49	0.60	0.50	0.60
Cloze	0.00	0.55	0.63	0.75	0.62	0.66	0.67	0.70	0.64	0.69	0.61	0.61	0.65	0.75	0.66	0.75
Decipher	0.00	0.14	0.00	0.00	0.24	0.00	0.07	0.44	0.00	0.41	0.19	0.07	0.15	0.44	0.17	0.44
Listening	0.00	0.49	0.17	0.27	0.42	0.34	0.31	0.35	0.33	0.45	0.37	0.38	0.35	0.49	0.36	0.49
O-Span	0.00	0.19	0.16	0.00	0.10	0.00	0.01	0.14	0.10	0.31	0.27	0.03	0.13	0.31	0.13	0.31
Symmetry	0.00	0.26	0.00	0.00	0.32	0.01	0.08	0.26	0.00	0.27	0.37	0.19	0.16	0.37	0.17	0.37
Verbal	0.00	0.41	0.53	0.17	0.13	0.14	0.08	0.17	0.34	0.38	0.47	0.36	0.28	0.53	0.	0.53
mean	0.00	0.36	0.30	0.25	0.33	0.24	0.23	0.37	0.27	0.43	0.38	0.30	0.31	0.50		
max	0.00	0.55	0.63	0.75	0.62	0.66	0.67	0.70	0.64	0.69	0.61	0.61	0.65	0.75		

Table 7. Accuracy of predictions of post-training improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	53.00	53.60	58.60	45.10	64.15	65.05	67.60	66.90	57.85	63.55	63.65	60.45	60.59	67.60
Listen	46.00	47.15	47.30	53.70	51.55	49.25	49.30	55.65	50.70	47.70	50.50	52.55	50.49	55.65
O-Span	54.00	65.75	63.95	58.35	66.20	61.10	54.90	62.10	68.80	63.05	66.75	67.90	63.53	68.80
Rotation	43.33	48.50	49.50	48.67	43.25	37.67	44.08	48.17	47.83	44.42	41.42	38.67	44.74	49.50
Symmetry	45.00	54.50	51.45	36.25	49.75	51.35	42.35	45.50	40.85	41.10	43.85	40.55	45.23	54.50
mean	48.27	53.90	54.16	48.41	54.98	52.88	51.65	55.66	53.21	51.96	53.23	52.02		
max	54.00	65.75	63.95	58.35	66.20	65.05	67.60	66.90	68.80	63.55	66.75	67.90		

Table 8. Accuracy of predictions of post-delay improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	62.50	59.75	79.05	52.80	70.60	80.80	81.95	75.50	56.80	56.90	77.70	71.55	69.40	81.95
Listen	61.50	60.15	44.65	65.80	57.55	59.15	59.20	51.05	52.55	58.15	55.00	55.10	56.21	65.80
O-Span	54.00	58.00	44.80	52.05	70.10	64.50	60.55	63.25	51.55	52.65	57.90	55.80	57.38	70.10
Rotation	53.33	50.42	43.92	45.00	46.25	47.92	46.75	43.75	44.92	50.58	45.50	45.42	46.40	50.58
Symmetry	62.50	55.95	67.60	50.20	68.15	73.55	71.90	79.25	63.00	61.00	71.65	67.35	66.33	79.25
mean	58.77	56.85	56.00	53.17	62.53	65.18	64.07	62.56	53.76	55.86	61.55	59.04		
max	62.50	60.15	79.05	65.80	70.60	80.80	81.95	79.25	63.00	61.00	77.70	71.55		

Table 9. Sensitivity of predictions of post-training improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	1.00	0.65	0.67	0.35	0.70	0.76	0.76	0.63	0.55	0.60	0.70	0.62	0.64	0.76
Listen	0.73	0.44	0.50	0.35	0.47	0.48	0.55	0.59	0.50	0.43	0.49	0.49	0.48	0.59
O-Span	0.00	0.55	0.64	0.43	0.62	0.60	0.52	0.61	0.54	0.60	0.61	0.57	0.57	0.64
Rotation	0.78	0.48	0.43	0.48	0.48	0.43	0.48	0.34	0.48	0.47	0.41	0.39	0.44	0.48
Symmetry	0.27	0.50	0.50	0.40	0.49	0.55	0.34	0.44	0.36	0.45	0.40	0.36	0.44	0.55
mean	0.56	0.52	0.55	0.40	0.55	0.56	0.53	0.52	0.49	0.51	0.52	0.48		
max	1.00	0.65	0.67	0.48	0.70	0.76	0.76	0.63	0.55	0.60	0.70	0.62		

Table 10. Sensitivity of predictions of post-delay improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	1.00	0.76	0.92	0.75	0.74	0.91	0.93	0.82	0.69	0.66	0.88	0.82	0.81	0.93
Listen	0.00	0.41	0.08	0.39	0.43	0.36	0.45	0.34	0.25	0.44	0.25	0.23	0.33	0.45
O-Span	0.00	0.46	0.42	0.29	0.69	0.58	0.55	0.60	0.37	0.41	0.46	0.42	0.47	0.69
Rotation	1.00	0.46	0.52	0.58	0.54	0.56	0.62	0.46	0.57	0.58	0.57	0.57	0.55	0.62
Symmetry	1.00	0.59	0.88	0.78	0.85	0.88	0.83	0.86	0.80	0.73	0.87	0.82	0.81	0.88
mean	0.60	0.53	0.56	0.56	0.65	0.66	0.68	0.61	0.54	0.56	0.60	0.57		
max	1.00	0.76	0.92	0.78	0.85	0.91	0.93	0.86	0.80	0.73	0.88	0.82		

Table 11. Specificity of predictions of post-training improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	0.00	0.41	0.49	0.56	0.58	0.52	0.58	0.71	0.61	0.67	0.57	0.59	0.57	0.71
Listen	0.18	0.50	0.45	0.73	0.56	0.50	0.44	0.53	0.51	0.53	0.52	0.56	0.53	0.73
O-Span	1.00	0.75	0.65	0.71	0.70	0.63	0.58	0.64	0.82	0.66	0.73	0.78	0.69	0.82
Rotation	0.11	0.49	0.55	0.49	0.39	0.33	0.40	0.63	0.47	0.43	0.43	0.39	0.46	0.63
Symmetry	0.61	0.60	0.53	0.33	0.50	0.49	0.51	0.47	0.46	0.38	0.48	0.46	0.47	0.60
mean	0.38	0.55	0.53	0.57	0.54	0.49	0.50	0.60	0.58	0.53	0.55	0.56		
max	1.00	0.75	0.65	0.73	0.70	0.63	0.58	0.71	0.82	0.67	0.73	0.78		

Table 12. Specificity of predictions of post-delay improvement by at least the amount of control subjects

	ZeroR	NB	BayesNet	IBk	Boost	Bagging	Ripper	C4.5	SVM	MLP	VoteMaj	VoteAvg	*mean*	*max*
Decipher	0.00	0.34	0.58	0.17	0.64	0.63	0.64	0.65	0.37	0.41	0.61	0.54	0.51	0.65
Listen	1.00	0.72	0.68	0.82	0.67	0.74	0.68	0.63	0.70	0.68	0.74	0.76	0.71	0.82
O-Span	1.00	0.68	0.48	0.72	0.71	0.70	0.65	0.66	0.64	0.63	0.68	0.68	0.66	0.72
Rotation	0.00	0.55	0.35	0.29	0.37	0.38	0.28	0.42	0.31	0.42	0.31	0.32	0.36	0.55
Symmetry	0.00	0.51	0.34	0.05	0.40	0.49	0.54	0.69	0.35	0.41	0.46	0.44	0.43	0.69
mean	0.40	0.56	0.49	0.41	0.56	0.59	0.56	0.61	0.47	0.51	0.56	0.54		
max	1.00	0.72	0.68	0.82	0.71	0.74	0.68	0.69	0.70	0.68	0.74	0.76		

Table 13. Top four features by Information Gain Ratio

Data set	Highest I.G. Ratio	2nd highest	3rd highest	4th highest
Decipher, post-training	Decipher1score	English1stLang	English2ndLang	Sex
Decipher, post-delay	Decipher1score	English1stLang	English2ndLang	Sex
Listen, post-training	English1stOr2nd	Migraine	English2ndLang	Sex
Listen, post-delay	Cloze1score	Education	English1stOr2nd	English1stLang
O-Span, post-training	Ospan1score	LangNumber	English1stOr2nd	Migraine
O-Span, post-delay	Ospan1score	English1stOr2nd	Education	English2ndLang
Rotation, post-training	English1stOr2nd	Sex	English1stLang	Education
Rotation, post-delay	Migraine	Sex	Education	English1stOr2nd
Symmetry, post-training	Migraine	English1stOr2nd	Education	Sex
Symmetry, post-delay	Ospan1score	English1stOr2nd	Migraine	Education

Table 14. Unordered sets of features selected by the CFS Subset algorithm

Data set	Features
Decipher, post-training	{ English1stLang, Decipher1score }
Decipher, post-delay	{ English1stLang, Decipher1score, Symmetry1score }
Listen, post-training	{ Sex, English1stOr2nd }
Listen, post-delay	{ Cloze1score }
O-Span, post-training	{ Education, Migraine, NumberLanguagesSpoken, Ospan1score }
O-Span, post-delay	{ Education, English1stOr2nd, Ospan1score }
Rotation, post-training	{ English1stOr2nd }
Rotation, post-delay	{ Sex, Education, Migraine }
Symmetry, post-training	{ Sex, Migraine }
Symmetry, post-delay	{ Sex, English1stOr2nd, Ospan1score }