Predicting Students' Learning Outcomes Using Eye-Tracking Data

Bertrand Schneider, Yuanyuan Pao, Roy Pea
Stanford University
schneibe@stanford.edu, ypao@stanford.edu, roypea@stanford.edu

ABSTRACT

In this paper, we describe how eye-tracking data can be used to predict students' learning scores. In a previous study [2], the first author conducted an experiment where dyads collaboratively worked on a set of contrasting cases. After performing a median split on the learning scores, we iteratively tried various machine-learning algorithms to classify students. We found that Support Vector Machine (SVM) with a quadratic kernel was able to correctly classify 93.18% of our test data using only rudimentary eye-tracking measures. We discuss how our results can be used to improve classrooms education.

1. INTRODUCTION

With the advent of cheap and widely available sensors, we are witnessing an alarmingly increase of data availability, what some call a ‘data deluge’, requiring new inquiry methods. This revolution is transforming virtually every research field. In biology for instance, there is so much data to process and understand that most computer science programs now have a specialization in computational biology. Education is only now coming to “big data” (recent developments include an Educational Data Mining Society, and for Learning Analytics – SOLAR), as researchers are collecting ‘in vivo’ data from thousands of students taking cognitive tutor based classes [9], and researchers collect more and more data on students taking Massive Open Online Courses (MOOCs); in both cases, data mining techniques can be advantageously used. In our paper, we focus on eye-tracking datasets collected from students.

2. THE CURRENT STUDY

In this section we describe the study that the first author conducted in order to gather our dataset. Please consult the following reference for additional information [19].

2.1 Methods

In this experiment, dyads remotely worked on contrasting cases (Schwartz & Bransford, 1998) to study how the human brain processes visual information. The experiment had three distinct steps: first, students were welcomed and assigned to two different rooms. They could collaborate via a microphone when working on the contrasting cases. In one condition, members of the dyads saw the gaze of their partner on the screen; in a control group, they did not have access to this information. They spent 15 minutes trying to predict how different lesions would affect the visual field of a human brain. They then read a text for another 15 minutes on the same topic describing how the visual pathways of the brain work. Finally, they individually took a learning test.

2.2 Measures

Our test measured learning on 3 distinct categories: memory, conceptual understanding and transfer question. We rated collaboration with Meier, Spada and Rummel’s [11] rating scheme. Finally, we categorized each participant as being a “follower” or a “leader” during the activity overall. We used the following indicators to categorize the members of the dyads: 1) who starts the discussion, 2) who speaks most, 3) who manages turn-taking in talk (e.g. “how do you understand this part of the diagram?”), and 4) who decides the next focus of attention.

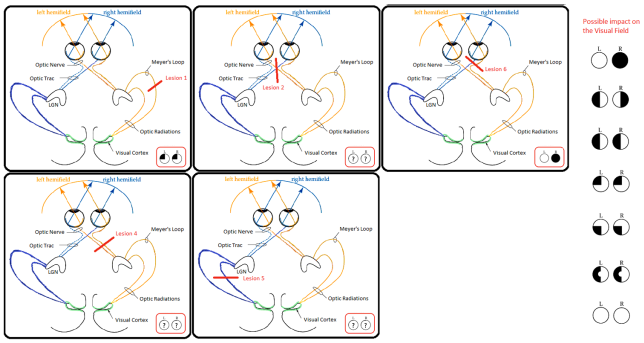

Figure 1: Contrasting cases used in this study. Subjects had the answer of two cases (top left and top right) and had to predict the results of a lesion on the three remaining cases.

2.3 Results

Results show that this intervention helped students achieve a higher quality of collaboration (F(1,10) = 24.68, $p < 0.001$) and a higher learning gain (F(1,40) = 7.81, $p < 0.01$). Additionally we categorized each member of the dyad as “leader” and “follower”. We found an interaction effect between those two factors (experimental conditions and individuals’ status) on the total learning score: F(1,38) = 5.29, $p < 0.05$. Followers learnt significantly more when they could see the gaze of the leader on the screen. They learnt less when they could not. Interestingly, participants in the “visible-gaze” condition achieved joint attention more often than the participants in the “no-gaze” condition: F(1,30) = 22.45, $p < 0.001$. The percentage of joint attention was one of the only measures correlated with a positive learning gain in our study: r = 0.39, $p < 0.05$. This result was confirmed by a mediation analysis (see [19] for more details).

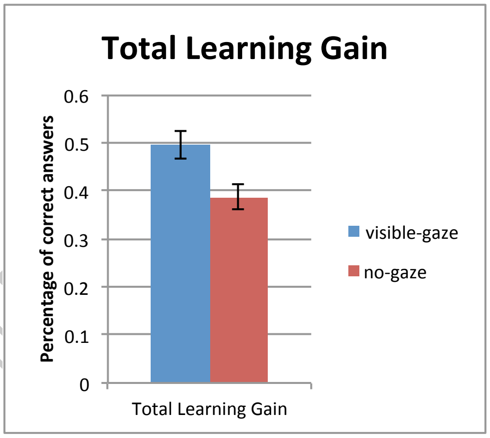

Figure 2: Results of the experiment. Dyads learnt more when using a "gaze-awareness" tool.

2.4 Eye-Tracking Data

The results of this study show the importance of supporting joint attention in collaborative learning activities. However, the eye-tracking data have been largely untouched in [19]. For each subject, we have the complete eye-tracking data for the first part of the study. We organize our data in the following way: first, we divided the screen into 7 areas of interest. This grid was defined to separate semantic regions on the screen. We then computed the number of fixation on each area (7 features) and the saccades between those regions (42 features). Finally we computed the minimum, maximum and average cognitive load of each subject based on their pupil size (3 features). In summary, we used 52 features to feed our learning algorithm. It should be noted that those features are very low level and can easily be computed from other eye-tracking data-sets.

3. Applying Machine Learning Techniques on Eye-tracking Data

Our goal is to find the best model to classify good and bad learners based on our gaze data. Since our entire feature set is significantly larger than our training set, we are very likely to over-fit our data and therefore perform poorly on new and unseen data. Our method, then, is to perform model selection and feature selection by trying various algorithms and combinations of the features. And because we want to maximize the number of training examples, we use hold-one validation to test our models and features.

Each of our features represents the counts of gazes per area of interest, the different saccades, or the cognitive load of the subject. We assume that these features can be treated as independent. Additionally, we choose to normalize it so that the relative magnitudes of the feature values can affect the model parameters. Then, given the normalized data and our independence assumption, we decided to apply three classification techniques: naïve Bayes, logistic regression, and support vector machines (SVM).

We first used our three algorithms by splitting the data in half and randomly labeling those two groups as "test" and "training" data. This simplistic approach led to poor results due to the small number of training examples. The performance of our algorithms greatly increased when we used the "leave-one-out" approach: we iteratively trained our algorithms on the entire dataset (minus one row) and predicted the category on this example. This process was repeated *m* times (where m = number of rows in our dataset).

3.1 Results from Cross Validation on Various Models

Table 1 shows the results that we obtained from running the three models on the gaze data set with and without feature selection. We describe how we performed feature selection in the following section.

Table 1. Accuracy from applying the three classification algorithms to our data. For SVM, a linear kernel was used by default (when not specified otherwise). Training accuracy is reported only when feature selection was not used.

		Naives Bayes	**Logistic Regression**	**Support Vector Machine (SVM)**
	Training	*86.58%*	*90.75%*	*100.00%*
	Test without feature selection	54.55%	63.67%	59.09%
	Test with f.s.	84.09%	52.27%	86.36%
Gaussian Kernel	**Test** with f.s.	*N/A*	*N/A*	84.09%
Polynomial Kernel	**Test** with f.s.	*N/A*	*N/A*	86.36%
Quadratic Kernel	**Test** with f.s.	*N/A*	*N/A*	**93.18%**

Both naïve Bayes and logistic regression performed poorly since they could not even perfectly fit the data that it was trained on. Even though the test accuracy from logistic regression seems to be better than SVM's test accuracy, if the model cannot even fit the training data with 0% error, then it is not capturing the right information from the training set.

Naïve Bayes did not succeed because it was trying to fit conditional probabilities on more than 75 features using only 43 training examples at a time. Because the discrete counts of our gaze data could range from anywhere between 0 and 21,600, there were at most one example per feature value in our conditional probabilities. As a result, we would not have valid conditional probabilities trained for values seen in the test example. Logistic regression suffers from the same problem; here, instead of finding conditional probabilities for each feature and each class, the algorithm was using only 43 examples to identify the proper weighting for more than 75 dimensions. In summary, there were not enough training examples to cover the entire feature space.

Our support vector machine model, using a linear kernel, proved to have the best performance on our data since it does not try to explain each data point. Instead, it tries to maximize the margin between the good and the bad classes in the training examples, and test points are classified based on their position relative to the identified boundary. Here, we get a 100% training accuracy but only a 60% test accuracy, which leads us to believe that our process is suffering from over-fitting. Therefore, our next step involved narrowing down our feature dimension using feature selection techniques.

3.2 Feature Selection

To solve our over-fitting problem, we tried to select the best combination of features to improve our performance. Unfortunately, our dataset had too many features for too few data points; for good accuracy, we actually only need the features that are the most indicative of the actual category. Here, we iteratively ran our best SVM model and added in features one at a time until we achieved our highest test accuracy. This "feature selection" technique is commonly used in machine learning to isolate the best set of features.

3.3 Final Results

After performing feature selection, we were able to achieve a test accuracy of 93.18% with a quadratic kernel (table 1), which is a substantial improvement from the 60% using a linear kernel without feature selection. This is a relatively impressive result considering that we are using only basic eye-tracking measures.

4. CONCLUSION

From using cross validation and feature selection on our models, we can see that, despite having very few training examples, we can create a very good classifier for learning quality. It is worth noticing that we can classify subjects' learning scores based solely on gaze data, without any more information about the subject (e.g., GPA, major, age). The accuracy of our classifying algorithm would probably increase with additional and more sophisticated features. Demographic data, speech data, and refined eye-tracking measures can advantageously be used to improve our results.

5. REFERENCES

[1] Koedinger, K.R., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2011). *A data repository for the EDM community: The PSLC DataShop*. In C. Romero, S. Ventura, M. Pechenizkiy, R.S.J.d. Baker (Eds.). Handbook of Educational Data Mining (pp. 43-55). Boca Raton, FL: CRC Press.

[2] Schneider, B., & Pea, R. (under review). Facilitating Joint Visual Attention with Gaze-Awareness Tools: An Empirical Study.*10th International Conference on Computer Supported Collaborative Learning*, CSCL2013. Madison, Wisconsin.