How well do people learn? Classifying the Quality of Learning Based on Gaze Data

Bertrand Schneider
Stanford University
schneibe@stanford.edu

Yuanyuan Pao
Stanford University
ypao@stanford.edu

ABSTRACT
In this paper, we describe how eye-tracking data can be used to predict students' learning scores. In a previous study, the first author collected eye-tracking data such as gaze position and pupil size while subjects either collaborated or worked independently on a problem. In this paper, we seek patterns in the eye-tracking data gathered during this experiment to accurately predict students' learning outcomes. We iteratively tried various machine-learning algorithms and found that a Support Vector Machine (SVM) with a quadratic kernel was able to correctly classify 93.18% of our test data using only aggregated eye-tracking counts. We then repeated this approach under a generalized setting where we extracted features after applying k-means clustering to the gaze data. The accuracy improved to 97.56%. These results show how machine-learning techniques can be applied to make qualitative sense of educational datasets.

1. INTRODUCTION

With the recent shift in education from the classroom to the web, there have been new questions arising as to how best to analyze the quality of learning. Learning analytics (LA) is defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs". Our project deals with "small" Learning Analytics, that is, using a large set of features over a small number of examples. Our goal is to predict students' learning gains when using this data as input features. From here, we can refine the range of measures and apply the resulting algorithm to larger sets of examples.

More generally, current work has been focusing on different levels of learning analytics: for instance, in text analytics, researchers have been trying to automatically classify the quality of an essay [2] and to create algorithms to help solve the problem of creating fast, effective and affordable solutions for automated grading of written work. Lastly, researchers have been trying to use artificial intelligence techniques to assess human learning.

1.1 Eye-Tracking and Gaze Analysis

Knowing the location of a user's gaze can provide insight into the user's visual attention and his eye-movement control mechanism. In education, this kind of measure can provide information on what students perceive and what they miss. Moreover, knowing where students focus their attention provides clues on their conceptual understanding of a phenomenon.

There has been some preliminary work on using machine-learning techniques on eye-tracking data. However, to our knowledge very few of them tried to directly predict learning outcomes.

2. EXPERIMENTAL SETUP

In this section we briefly describe the study that the first author conducted in order to gather our dataset.

2.1 Methods

The experiment had three distinct steps: first, students were assigned to different rooms. They could collaborate via a microphone when working on a set of contrasting cases. In one condition, subjects saw the gaze of their partner on the screen; in a control group, they did not. They spent 15 minutes trying to predict how different lesions would affect the visual field of a human brain given a document describing how lesions affect the brain as well as the diagrams shown in Figure 1. (Note that we do not use the speech data and have treated all of the subjects as independent.)

2.2 Measures

At the end of the learning period, each subject took a learning test to assess their understanding of the topic. Our test measured learning on 3 distinct categories: memory, conceptual understanding and transfer question. We rated collaboration with Meier, Spada and Rummel's [1] rating scheme. For this first attempt, we only tried to predict learning scores. In

future work we will also try to predict the quality of collaboration based on the eye-tracking data.

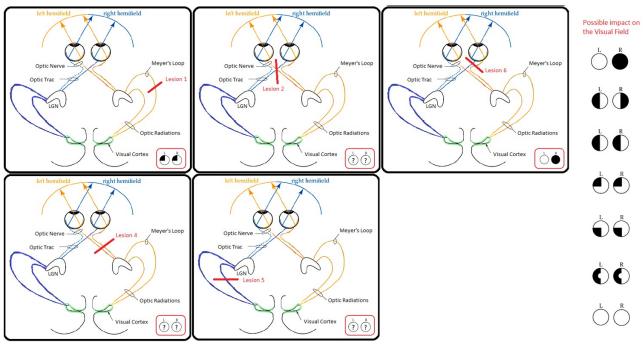

Figure 1: Contrasting cases used in this study. Subjects had the answer of two cases (top left and top right) and had to predict the results of a lesion on the three remaining cases.

2.3 Eye-Tracking Data

For each subject, we have the complete eye-tracking data for the first part of the study: this means that the user's gaze was captured 30 times per second. This resulted in approximately 50,000 data points for each participant (a data point can describe the location of either a fixation or a transition or the size of the participant's pupil at certain time point). In total, we have 44 subjects * 50,000 measures = 2,200,000 data points that we can exploit as input features for our machine learning algorithm.

2.4 Manual Feature Extraction

We organize our data in the following way: first, we divided the screen into 7 areas of interest based on the diagram shown in Figure 1. This grid was defined to separate semantic regions on the screen. We then computed the counts of fixation on each area (7 features) and the transitions between those regions (49 features). Finally we computed the minimum, maximum and average pupil size for each example. In summary, we aggregated the raw gaze data into 56 features to feed into our learning algorithm.

3. CLASSIFICATION VIA MACHINE LEARNING

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 "leave-one-out" validation to test our models and features.

3.1 Model Selection

Each of our features represents the counts of gazes per area of interest, the different transitions between areas, or the cognitive load (pupil size) of the subject. We assume that these features can be treated as independent. Additionally, we choose to normalize it so that the relative magnitudes among different features cannot 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).

3.2 Cross Validation

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 and test examples. We used the "leave-one-out" approach to obtain more reliable test and training errors: 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).

One advantage of SVMs over other techniques is the ability to work in a high-dimensional feature space by using kernels. During model selection, we also varied our SVM algorithm by using different kinds of kernels (linear, quadratic, Gaussian and polynomial).

3.3 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 needed 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.

3.4 Results

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%
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%
Quadratic Kernel using K-Means Segmentation	**Test** with f.s	*N/A*	*N/A*	**97.56%**

Both naïve Bayes and logistic regression performed poorly because 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.

For Naive Bayes and logistic regression, there were not enough training examples to cover the entirety of our high-dimensional feature space. Naïve Bayes did not succeed because it was trying to fit conditional probabilities on more features than training examples, so, in many cases, there were at most one example per feature value and, as a result during testing, would either not have seen the values or over-fit for them. Logistic regression suffers from a similar problem except, here, the algorithm was using a few examples to identify the parameters for many more dimensions.

Our support vector machine model, when using a default linear kernel, proved to have the best performance on our data. Since it does not try to explain each data point (like logistic regression and naïve Bayes) but instead tries to maximize the margin between the two classes in the training examples, we achieved a 100% training accuracy but only a 59.09% test accuracy, which led us to believe that our process was suffering from over-fitting and required feature selection.

After performing feature selection, we were able to achieve a test accuracy of 86.36% with a linear kernel and 93.18% with a quadratic kernel (Table 1), which is a substantial improvement from the 60% test error when using a linear kernel without feature selection.

4. GENERALIZATION WITH K-MEANS

For our particular data set, it is easy to reduce an individual's gaze data points into counts per region since we knew exactly what was displayed on the screen and which regions were important towards learning. Now, we want to see how our classifiers perform if we assume only the raw gaze data and no extra information about the problem.

Because each individual subject has a lot of raw gaze data points, we cannot avoid the task of aggregating counts over regions of the screen, just like in Section 2.4. This is where we apply k-means over all of the available gaze data points to identify the k clusters in our problem. We treat these clusters as the regions and feed the counts data as the features for our classifier, just as before.

4.1 Choosing a Value for k

In preparing the data, we first needed to decide on the value for k. Again, we applied SVM to the resulting data for different values of k and chose the value that yielded the highest leave-one-out cross validation accuracy.

We tested for values of k from 1 to 30. If the training accuracy was not 100%, we did not even look at its test accuracy. Also note that, for certain values of k, k-means would not converge. A plot of the resulting accuracy versus k is shown in Figure 2.

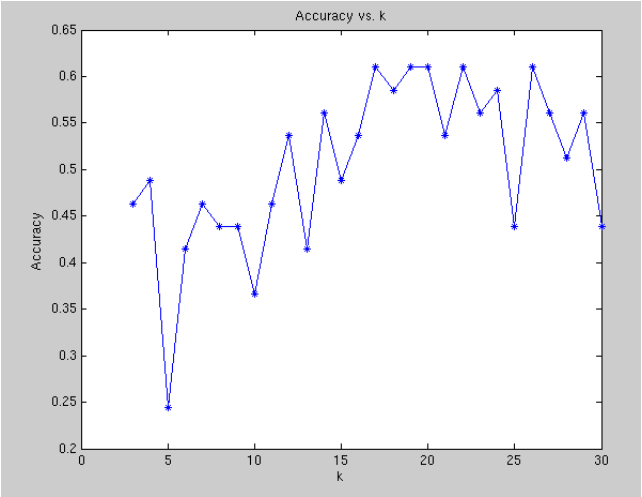

Figure 2: Parameter Tuning for k: Test accuracy (using LOOCV) versus the value of k as it is varied from 1 to 30.

As expected, the accuracy is much lower for smaller values of k. For the values of k where k-means did not converge, the accuracy values were not consistent and, therefore, unreliable, even though they are depicted in Figure 2. Therefore, the best value of k that converged was k=19 with an accuracy of 60.98%, and the centroids are marked on the diagram in Figure 3.

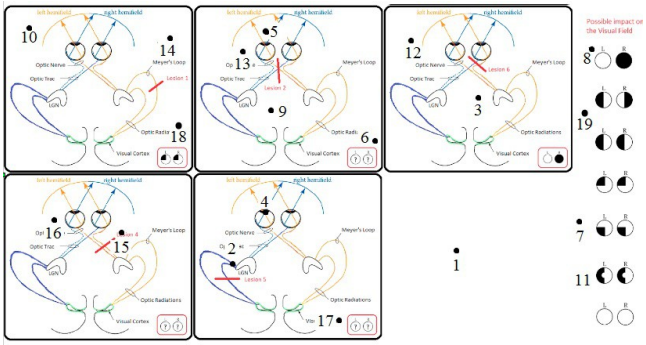

Figure 3: Cluster Centroids: The 19 centroids identified through k-means are marked on the screenshot of the problem that our subjects were working on.

4.2 Feature Selection

With the counts of gazes per 19 clusters and transitions between those clusters, we have the input data for our SVM classifier, but we want to be able to do much better than the 60.98% LOOCV accuracy achieved when we use every feature. As before, we want to perform feature selection to find the best combination of features for classifying subjects as good or bad learners. The results of running feature selection are shown in Figure 3.

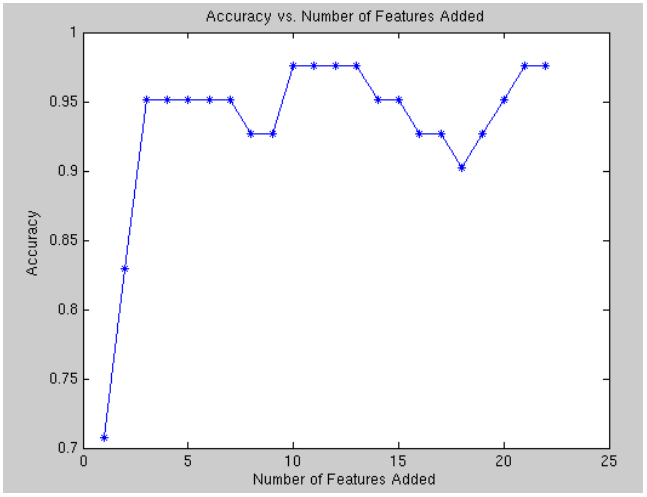

Figure 4: Results from Feature Selection: Our classifier improved greatly with the first three features added, and our best result was reached with just ten features.

With just one feature, our performance was already better than when we used all of the data. And with just three features, we already achieved an accuracy of 95.12%, which is better than that of this exact classifier with the manually defined clusters described in Section 2.4. Our highest accuracy was 97.56%, which first occurred with ten features.

The top ten features identified through feature selection and yielded the best performance are listed in Table 2: (in order of output)

Table 2. Feature Selection Results: The top ten features that are sufficient for a 97.56% LOOCV accuracy on our data set.

Output Order #	Feature Name	Output Order #	Feature Name
1	16 → 6	6	5 → 17
2	11 → 2	7	5 → 15
3	1 → 16	8	17 → 5
4	15 → 8	9	14 → 7
5	10 → 17	10	5 → 1

It is also interesting to note that gaze counts for each cluster is not among the top ten features nor the top thirty. The only features that were most useful were the number of transitions between certain clusters.

The particular transitions in Table 2 are technically the most indicative of understanding, so we can possibly infer which pairs of locations on the screen are good or bad connections. If we cross reference these features with the centroids shown in Figure 3, we can see that some of these features involve

transitions from a related answer choice on the right of the screen to a position on a diagram with a related lesion depicted. For other features, we can see transitions between unrelated points, which could be a feature that helps classify bad learners. These top features provide insight into good and bad approaches to finding the correct answer to the problems.

4.3 Multiple Values of k

We also tried consolidating the count data from multiple values of k as the input features to our SVM classifier to see if we could improve our accuracy, but when we ran feature selection, we noticed that we were picking out only the features associated with the highest value of k (that converged) to get the best performance. Therefore, we resorted to using only one k value at a time.

5. CONCLUSION

Our preliminary results show that very rudimentary eye-tracking counts can accurately predict complex outcomes such a student's learning or the features that most indicate good or bad learning. These findings provide exciting perspectives for online and in-situ education: analyzing gaze movements can provide a better assessment of understanding. Even under generalized settings where we know nothing about the problem that is being solved, we can use the gaze data to identify regions and then aggregate count data for our classifiers. The performance in this general case was 97.56%, a 4.38% increase from when we manually defined those regions.

In addition, the top features identified via feature selection were extremely helpful in providing insight into the problem's characteristics: a correct answer and its supporting diagram, uninformative regions, etc. We can see that the features chosen, together with the locations of the centroids from k-means, can help indicate what places were more confusing or helpful to the subjects.

More data will allow us to make even more fined-grained predictions like predicting whether a particular misconception is likely to arise among particular students. From another perspective, we can also train machine-learning algorithms to separate students who answer a test by using rote memorization or critical thinking. This is a particularly valuable approach considering the difficulty of assessing students' thinking skills and the focus that is currently put on students' "21th century competencies".

6. FURTHER WORK

Our classification performance was remarkably high, and this is partly a result of having very few data points. We would like to gather more data for different types of problems and different types of learning styles to be able to evaluate where our methods are weaker and require more improvement.

Since the results from Section 4 demonstrate that we can find connections between clusters via feature selection, another next step would be to evaluate the accuracy of this method. We would like to separate these features by which category the support falls under. Knowing which features are associated with a good learning approach can help struggling students establish the right connections – for example, if their gaze is at a particular point, then we can suggest the next location they should be looking at and see if this scaffolding helps. Overall, our results lead to more questions and more insights into education that help tailor studying to each student's learning style.

More generally, we envision our results being applied to a multitude of learning situations as eye-tracking devices become cheaper and widely available. Not only would this approach provides more precise means of assessing students' learning, but also give teachers a direct and potentially real-time feedback on the kind of concepts that students struggle with.

7. REFERENCES

[1] Meier, A. et al. 2007. A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning*. 2, 1 (Feb. 2007), 63–86.

[2] Pellegrino, J.W. and Hilton, M.L. 2012. Education for life and work: developing transferable knowledge and skills in the 21st century. *National Research Council, Washington, DC*. (2012).