Goals of the Project

The goal of this project is to determine if we can use eye-tracking data to determine the expertise of the developer. We also want to see if we can use eye-tracking data to determine the difficulty of a coding task. If we can predict how difficult a task is we might be able to determine when a developer needs to take a break to avoid unnecessary bugs. If we can predict expertise, it helps in automatically characterizing experts in an objective way instead of relying on a questionnaire that they fill out.

Process

We conducted two experiments related to our goals.

First experiment: After extensive literature review, we decided to conduct an eye-tracking study to determine if novices and experts solve algorithm related problems differently. We gathered a list of topics that were covered in the Fall 2014 algorithms analysis course at Youngstown State University (YSU). We designed tasks for our subjects that resembled problems from the homework or from the tests in that course. At the end of the Fall 2014 semester, we gathered eye-tracking data for four participants (3 novices and 1 expert). The three novices were students enrolled in the course that fall and the expert was the professor who taught the course.

We gathered the data in the Software Engineering Research and Empirical Studies Lab (SERESL) at YSU. All participants had to complete a pre-questionnaire, post-questionnaire, and NASA TLX surveys for each task. They all were given as much time as they needed to solve all eight tasks. Each participant solved the same eight tasks as the others albeit in a randomized order. We tracked their eyes while they were solving the tasks.
Second experiment: In this experiment, we used existing eye tracking data sets that had been collected in the eye tracking lab to see if we could apply our machine learning algorithms on them. The first step that needed to be done was to get the data into a format that the machine language algorithms could understand. This data was already categorized into expert and novice samples and the tasks were also categorized as difficult or easy.

Our goal is to use data mining algorithms to predict expertise and task difficulty. These are two binary classification tasks. To be able to predict expertise and task difficulty we combined all our data in one flat file that contains more than three million instances. Data mining tools like Weka would not work very well due to the size of the file. We started learning the Machine Learning Studio available on Microsoft Azure. We uploaded the data and we plan to run Naïve Bayes, Decision Trees and Support Vector Machines as the mining algorithms.

Conclusions and Results

First experiment results: Following the gathering of new data, we began to analyze the data that we gathered and the data that we had compiled previously. The study we performed did not contain enough data to use with machine learning, so we performed a different analysis as presented below on the data.

With the data obtained from the study we looked at several different issues. We looked at correctness, time taken for each task, fixation counts, fixation durations, and transition matrices. From the time taken for each task we can calculate the overall time to complete all tasks. From the data received from the eye tracker we analyzed fixation counts and fixation durations. We were also able to generate a transition matrix for each participant from each task from the raw data obtained from the eye tracker. This gives us a visual representation of how the participants look at the different areas and how many times they transition between them. Novices took much longer than the expert. Novices also had many more fixations than experts. Based on the analysis we can see differences in the way that experts and novices solve problems. However, in order to see if this is based on the tasks being solved or a generalization of all tasks more data is needed to run machine learning algorithms.

Second experiment results: When the data file was uploaded in the cloud many of our numeric features imported as string. Before we can run any prediction tasks we need to preprocess the data and transform the features to the correct data type. Only certain columns are selected and the data needs to be split into two sets: training and testing. Our plan is to continue to work on this part of the project during the summer.

Summary: Our goal was to be able to automatically predict expertise and task difficulty. In order to start to provide some results in this area, we gathered some eye tracking samples of an algorithms class in Fall of 2014. It involved designing and conducting a small pilot study on algorithm related problems. This study helped us in refining many aspects of the data collection. We plan on running this study every Fall when the algorithms class is offered. While looking at the measures in eye tracking data that we collected, we noticed some significant differences in the way the expert solved the problem vs. novices. The expert would zone into very specific places and answer much quicker than a novice. Novices pretty much looked all over the place because they did not find any clue to solve the problem. This finding will directly help us in our mining algorithm choice of features when we run it in the summer. We also think these results can help an instructor to see how they can improve the curriculum thereby making it easier for students to learn algorithms (a difficult class for many students).

Presentations and Publications


Whitely, J., Wise, J., Lazar, A., Sharif, B., “Towards Understanding Student Problem Solving Behavior via Eye Tracking” at the Annual Youngstown State University QUEST: A forum for student scholarship, April 7, 2015, accepted as an oral presentation (http://www.ysu.edu/QUEST/)