GAZE ASSISTED PREDICTION OF TASK DIFFICULTY LEVEL AND USER

ACTIVITIES IN AN INTELLIGENT TUTORING SYSTEM (ITS)

A Thesis

by

PURNENDU KAUL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,   Tracy Hammond
Committee Members,   Anxiao Jiang
                                  Daniel W. Goldberg
Head of Department,   Dilma Di Silva

May 2016

Major Subject: Computer Science

ABSTRACT

Efforts toward modernizing education are emphasizing the adoption of Intelligent Tutoring Systems (ITS) to complement conventional teaching methodologies. Intelligent tutoring systems empower instructors to make teaching more engaging by providing a platform to tutor, deliver learning material, and to assess students' progress. Despite the advantages, existing intelligent tutoring systems do not automatically assess how students engage in problem solving? How do they perceive various activities, while solving a problem? and How much time they spend on each discrete activity leading to the solution?

In this research, we present an eye tracking framework that can assess how eye movements manifest students' perceived activities and overall engagement in a sketch based Intelligent tutoring system, "Mechanix." Mechanix guides students in solving truss problems by supporting user initiated feedback. Through an evaluation involving 21 participants, we show the potential of leveraging eye movement data to recognize students' perceived activities, "reading, gazing at an image, and problem solving," with an accuracy of 97.12%. We are also able to leverage the user gaze data to classify problems being solved by students as difficult, medium, or hard with an accuracy of more than 80%. In this process, we also identify the key features of eye movement data, and discuss how and why these features vary across different activities.

Dedicated to Punam Kaul, Subash Kaul and RaghuNath Kaul

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Interactive systems including modern day intelligent tutoring systems(ITS) like Mechanix [32, 2, 43, 44, 16, 42, 18], iCanDraw [13, 11], ITS for East Asian language learning [38, 39, 37, 40, 36], biology ITS [41], or child-based ITS [26, 24, 25, 45], mostly provide a user-initiated feedback mechanism. This is because of the inability of the system to know the user's cognitive state, behaviour and intent in real time. Modelling human behaviour can prove to be very useful in developing adaptive interfaces for such systems. The input user information for such systems has to be collected using methods which are unobtrusive but sensitive and fast to ensure a real-time and accurate response. Eye tracking technology can fit this role very well because it provides a real-time behavioural index of the visual and cognitive processes.

Gaze movement data is a promising input for modeling cognitive and perceptual processes because the gaze response to visual stimuli is regular and systematic(as researched by Yarbus[47] and Rayner[34]). Taking cue from the eye-mind hypothesis given by Just and Carpenter[22] many studies have now been done to analyze the relation between gaze and human cognition process. In the research done for this thesis, we have investigated the possibility of using eye gaze data to classify problems being solved by students on an ITS as difficult, medium, or hard. We also classify the activity being performed by users as "reading", "gazing at an image", and "drawing/problem solving." Knowing the level of difficulty being perceived by the user can help the system adapt it's interface accordingly and assist the user better. There can be scenarios in which the users interact with content which is not static (like reading material, problems etc already in an ITS) but produced by

another human in real time (dynamic). A good example of such a scenario is when multiple users use the digital whiteboard on devices like Microsoft Hub to learn or communicate ideas. In such conditions eye movements of users can be used to know the activity they are doing(reading a handwritten text, gazing an image drawn by others, or problem-solving).

We have analysed gaze data of users which were collected while they solve problems on Mechanix [23, 1, 23, 33]. Mechanix is a sketch-based tutoring system for engineering and high school students enrolled in statics based courses. It allows students to submit hand-drawn solutions to static problems with planar truss and free body diagrams, just as they would with pencil and paper problems. Mechanix checks the students work against a hand-drawn answer key provided by the instructor. Figure 1.1 shows the Mechanix interface. Each problem displayed on the interface consists of a problem statement, an image on the right, control panel which consists of the tool(pencil and eraser) and action(undo, delete) buttons, a drawing area, an equation panel and a status panel to the left of the drawing panel. The users draw the truss, input forces and reactive forces in the drawing area. Each time the user clicks the 'CORRECT' button in the control panel the system checks the user-input solution and gives feedback. The status panel also gets updated at this time to show parts of the solution which are incomplete/complete.

of 2

A stationary block or truss does not rotate, therefore the moments about each node must sum to zero. To compute the moment about a node, multiply each external force by the distance of the force from that node. For example if B is centered between A and C as pictured, with a distance of x between each node, then the sum of moments about A=0 gives us: x*(-10) + 2x*Rcy = 0. (This is treating counter-clockwise moments as positive.). In the adjacent diagram, node C is a "roller node". Therefore there will be no Rcx. Do not label it.

ACTIONS   TOOLS   CORRECT?

Unable to detect an input force at node B. You might need to redraw. Sorry !

Question 1

Set up the problem

Input Forces

Reaction Forces

D

A   B   C

Ray

T

HIDE EQUATION PANEL

Equations: ΣFx = 0 =
             ΣFy = 0 =
             ΣM   = 0 =

Ray ???   (units) ▾

Figure 1.1: Mechanix interface

2. RELATED WORK

2.1 Modeling user behaviour, cognition and intent

Extensive prior research has shown the effectiveness of eye-tracking in monitoring cognitive processes. This includes research aimed at differentiating reading from non-reading activities [8], [12]. Eivazi and Bednarik [4] trained a support vector machine to classify high-level behavioral codes by using features computed from the eye movement data collected while subjects solved an 8-tile puzzle. The prediction model developed by them also gives a 66 percent accuracy in classifying users into three categories based on individual performance in solving puzzles: high, medium and low. Eivazi et al. claim that the model prediction increases to 87 percent when there are only two classes. Several researchers have worked to use the gaze pattern of users to study and predict reading and comprehension skills of the users. Navalpakkam, Rao, and Slaney [31] use the gaze pattern to predict the reading struggles of users due to distraction. Gomez and Aizawa in their comprehensive study on reading behaviour [30] use a combination of eye-movement features and characteristics of the document to predict the language skill and level of understanding of the participants. Kunze et al. [28], who have also used eye tracking to assess a user's reading skill level, devised a method to spot difficult words for a user. Their experiment was a paper-based experiment and they laid emphasis on the regression feature to narrow down their search for difficult words in a reading assignment. Jiang et al. [21] examined the pupils' response to the difficulty level of a task through an experiment in which the task requirements were managed by the Fitt's index [17] of difficulty. This study showed that the positive change in the pupil diameter is abrupt for higher task difficulties and that the peak pupil dilation occurred after

a long delay at the end of task completion. Elmadani et al. [15] in their work on studying the effectiveness of tutorial dialogues in an intelligent tutoring system take the help of eye movement data but only consider a very limited and superficial set eye movement features. Chen, Epps and Chen [9] have used pupil action, gaze and blink data to infer task characteristics by estimating levels of perceptual and cognitive load. They use a set of about 29 abstract features each calculated in a time window of 1 or 2 seconds. Researchers have tried to model human reading and differentiate it from closely related activities like skimming [7], [14]. Studies by Simola et al. [35] show that a model based on hidden Markov models(HMM) can achieve an accuracy of 60.2% in predicting the task a user is engaged in out of the three information search tasks: word search, searching for an answer and choosing an interesting title. Bednarik et al. tackle the Midas Touch problem [20] of intent prediction in gaze-based interaction by using SVM and report an accuracy of 76% [5].

2.2   Gaze based interactive Applications

Eye-tracking has also been used extensively for application research. Xu [46] and Loboda [29] show that eye movements of users while reading can be used to summarize documents or find relevant words. Text 2.0 framework developed by Biedert et al. [6] enables writing web-based gaze-controlled real time applications. The framework was used by them to write a plug-in Text 2.0 which enables text responsive reading experience. It assists the user to overcome comprehension difficulties when reading. It does not do any behavior or cognition modeling but is a very good example of using eye-tracking for real-time interaction with systems.

3. EXPERIMENT DESIGN AND PROCEDURE

In order to investigate the use of eye gaze data for classifying on-screen activities and identifying tasks on the basis of difficulty level, we designed an experiment to gather eye-movement data of users while they work on a tutoring system. We wanted to engage users in tasks which have varying levels of difficulty and comprise of different activities (reading, gazing image and drawing/typing). At the heart of the experiment is Mechanix [32], which is a sketch-based intelligent tutoring system that allows students to hand-draw solutions to truss problems just as they would with pencil and paper. It then checks the students' work against a hand-drawn answer entered by the instructor.

The Figure 3.1 shows the Mechanix interface. This experiment engages the user in broadly three different activities namely reading, drawing/typing and gazing while they solve truss problems. All users attempt to solve three assignments each having two 'truss' based questions. The difficulty level of the assignments increases from easy to medium and finally hard as we go from the first to the third assignment. The interface is divided into three main panels. The reading panel (r_panel) at the top has the problem statement. Each problem statement is complemented by an image on the right-hand side, inside the image panel(i_panel). In the center is a drawing panel(d_panel) inside which the students draw the truss and label the input and reactive forces. We modified the Mechanix code to incorporate an eye-tracking system which tracks and records the eye movement of the user as they solve the assignments and then saves this data for each assignment separately. The eye movements are recorded using Eyetribe which is placed at the bottom of a 21 inches Cintiq display (resolution: 1600x1200) at a distance of 60 cm from the user. The

users used a mouse to draw the truss diagrams. Twenty-one participants in the age group 22-30 years, solved three assignments. All the users were given a practice assignment before the experiment to familiarize them with the interface and the nature of the tasks. The users labeled each assignment by its difficulty level. By the end, we had gaze-data from 63 user studies (21 for each easy, medium and hard assignments). R and Weka were used to do data analysis.
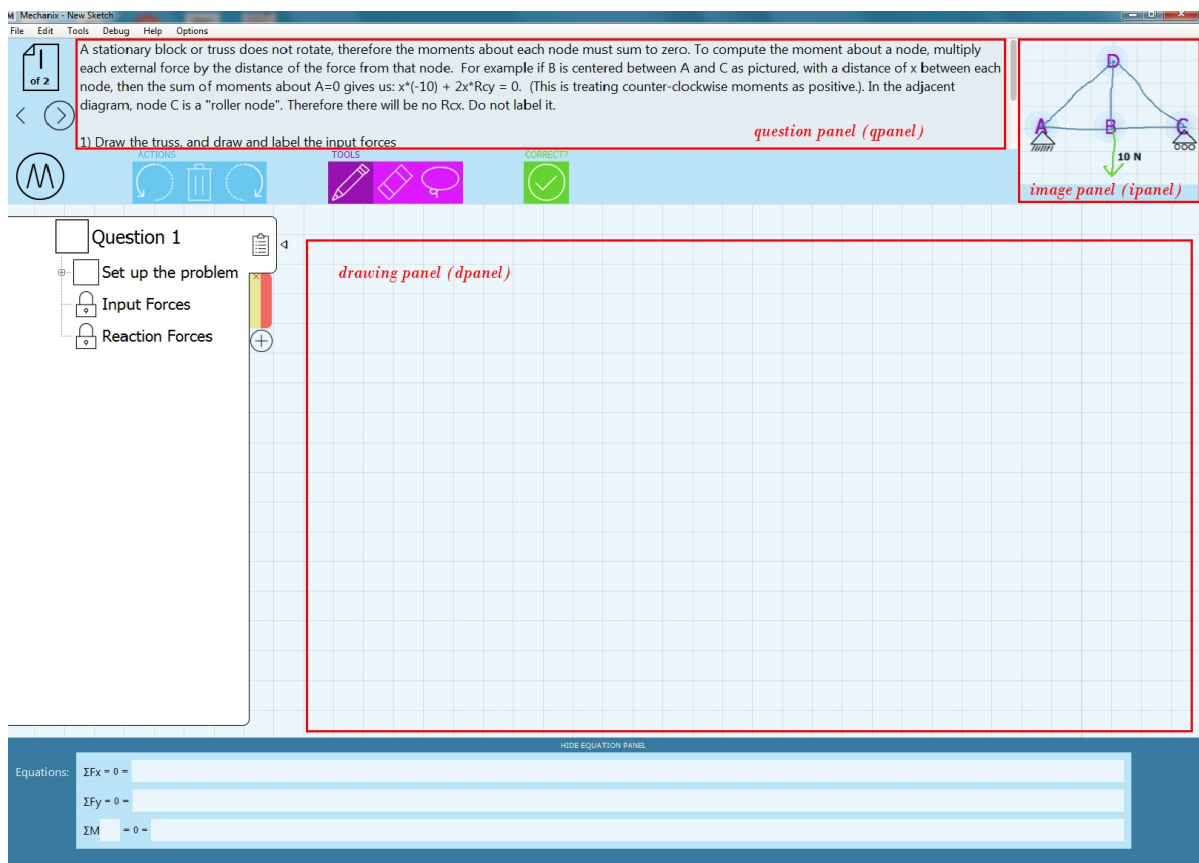
Figure 3.1: Interface of mechanix

4. DATA ANALYSIS

We follow a simple pipeline to analyse the data and build a classification model for both the problems. This pipeline broadly consists of five stages (Figure 4.1). Some of these stages may be significant for only one of the two problems.

Data Cleaning and Pre-processing
⇩
Feature calculation and Exploratory Analysis
⇩
Feature Selection
⇩
Dimensionality Reduction
⇩
Classification model

Figure 4.1: Analysis pipeline

1. **Data Cleaning and Pre-processing**

   The eye tracker collects data at a frequency of about 30 Hertz. Each data point corresponds to a different time-stamp and eight other values. Figure 4.2

shows a snapshot of raw data. The eye tracker labels each of these data points with a 'state'. The state column can have any of the four states are shown in the Table 4.1. State '7' represents the perfect state in which the eye-tracker is able to track the eyes perfectly and returns the gaze coordinates and the pupil diameter for both eyes. State '4' is also considered a valid state. However, the eye tracker may miss out on reporting one of the four values(SmoothX, SmoothY, RightEyePupil and LeftEyePupil) for such data points.

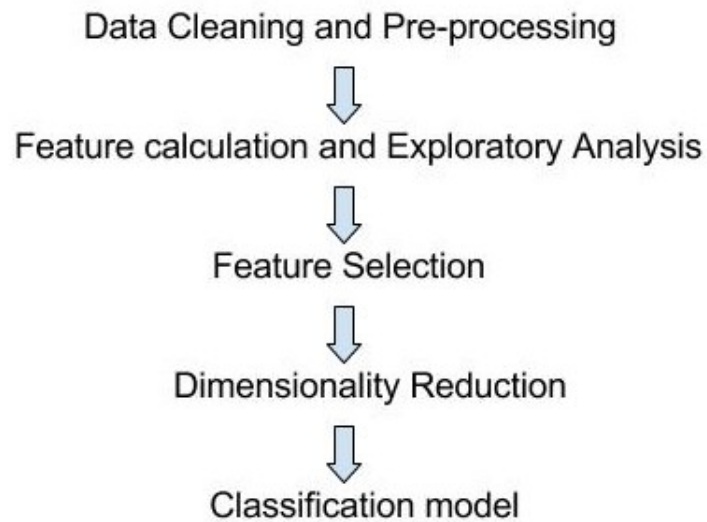|	A	B	C	D	E	F	G	H
1	State	StateToString	TimeStamp	SmoothX	SmoothY	IsFixated	LeftEyePupil	RightEyePupil
2	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.390	882.6237	210.3226	false	16.9124	15.895
3	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.423	685.1874	207.6915	false	17.6438	15.4874
4	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.456	671.8745	189.7611	false	17.492	15.6322
5	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.490	679.2203	197.1102	false	16.0607	14.5976
6	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.523	679.978	188.577	false	15.4018	13.8861
7	7	STATE_TRACKING_GAZE | STATE_TRACKING_EYES | STATE_TRACKING_PRESENCE	2016-01-06 14:23:08.556	677.3297	191.5955	false	16.1799	13.6994

Figure 4.2: Raw data sample

State	String	valid
4	TRACKING_PRESENCE	Yes
7	TRACKING_GAZE TRACKING_EYES TRACKING_PRESENCE	Yes
8	TRACKING_FAIL	No
16	TRACKING_LOST	No

Table 4.1: States of gaze points from eyetribe

Points in state seven or four only are considered valid points. The user studies which had less than seventy percent valid data points were not used in this study. This left us with 58 user studies(17-easy, 21-diff, 20-med). Each data point carries a property 'IsFixated'. This property labels the data point as a 'fixation point'(TRUE) or a 'saccade point'(FALSE). Other columns are the 'TimeStamp', 'Smoothx', 'SmoothY', 'LeftEyePupil' and 'RightEyePupil'. SmoothX and SmoothY correspond to the average X and Y coordinate of gaze position on the screen and the Left/Right-EyePupil give the diameter of the left and the right eye pupil respectively. The raw data was cleaned to get rid of the invalid data points before the features were calculated. We also categorize the data points by labeling each of them with a value for the variable 'panel'. The screen was divided into three primary panels each of which represents a different activity. The table 4.2 lists the three panels shown in Figure 1.1 and the activities associated with each one of these panels.

Panel	Activity
r_panel	Reading text
d_panel	Drawing and problem solving
i_panel	Gazing at an image

Table 4.2: Panels on the screen and related activity

2. **Feature calculation and Exploratory Analysis**

We used R for calculating features and doing an exploratory analysis on gaze data. A lot of different features can be calculated from the raw gaze data. We were mostly interested in features which are not dependent on the duration of an assignment. This makes our study independent of the duration for which a user uses the system. However, as part of the exploratory analysis, we have calculated some features that do not follow this rule.

The following two definitions help us understand how gaze points for feature calculation were selected.

(a) **Fixation set** is a set of edges between consecutive fixation points. A sample gaze data may contain several such sets. In the Figure 4.3 there are two fixation sets: {1,2,3,4,5,6} and {13,14,15,16}. Gaze point 8 has no adjacent fixation points and hence no is not contained inside any fixation set. Each edge in the set has a length and time interval associated with it. The time interval is the time taken to traverse the edge. For each fixation set i we can calculate fixation_distance FD_i which is a sum of all edge lengths, fixation_time FT_i which is the sum of all time intervals and the total number of edges:FE_i.

(b) **Saccade set** is a set of edges between consecutive saccade points. A

11

sample gaze data may contain several such sets. In the Figure 4.3 there is only one saccade set: {9,10,11,12}. Saccade point 7 has no adjacent saccade point and hence no saccade set associated with it. Similar to a fixation set, each saccade set i has a saccade_distance SD_i which is the sum of all edge lengths, saccade_time ST_i which is the sum of all time intervals and the total number of edges:SE_i.
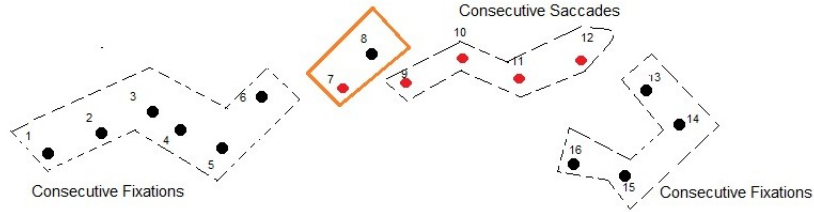
Figure 4.3: Sample gaze data

Mean path distance is defined as the average distance between two consecutive fixation points. We find all the fixation sets in a data set. For a sample data with n such sets from 1 to n, mean path distance is given by the following equation:

$$MeanPathDistance = \frac{\sum_{i=1}^{n} FD_i}{\sum_{i=1}^{n} FE_i}$$

FixationSpeedX is the average speed of gaze between fixation points in the x-direction only. For each fixation set i we can calculate FDx_i which is the

sum of the absolute change in x-coordinates for consecutive fixation points. Average speed is given by the equation:

$$FixationSpeedX = \frac{\sum_{i=1}^{n} FDx_i}{\sum_{i=1}^{n} FT_i}$$

Other features like Mean Saccade Length, SaccadeSpeedX and SaccadeSpeedY are calculated using the similar definitions. We also looked into the features which can be calculated from the pupil activity. Jiang et.al[21] have shown that pupil diameter can be used as indicators of task requirements in goal-directed movements. For this analysis, we were only interested in the features which are independent of the duration of the task. Dilation speed is simply defined as the speed with which the pupil dilates(expands or contracts). Saccade-related features have also been used by researchers like Gomez and Aizawa[30] to classify the understanding level and language skill of students.

3. **Feature Selection**

We use the Caret[27] R package to help us in the feature selection process. Data can contain features which are highly correlated (redundant). Several classification methods work well if redundant features are left out. Therefore, the highly correlated variables are identified and dropped from further analysis. We remove the attributes with an absolute correlation of 0.75 or higher.The second step is to estimate the importance of each feature by building a model and evaluate the importance of each feature by its performance in that model. We build a Learning Vector Quantization(LVQ) model and then use the varImp

function in Caret package to evaluate and report the importance of each feature. Finally, we use the Recursive Feature Selection(rfe) method to choose the best set of features. Reduced feature elimination helps us maintain a balance between accuracy and the number of features to be used.

4. **Dimensionality Reduction**

We may have to reduce the dimensionality of the data to avoid over-fitting. This is particularly true for this study because of the low number of data points. Therefore, once we have the best set of features which describe the data we reduce the dimensionality not just by picking the top features but by doing a Principal Component Analysis(PCA) on the variables to get the top three or four components to be used for the classification task. Reducing the dimensionality of the data also helps us visualize the data better.

5. **Classification model**

The last part is to test the performance of various classification algorithms. We report the accuracy and f-measure along with the confusion matrix for the best performing classifier.

5. RESULTS AND DISCUSSION

In this chapter we discuss the results of exploratory analysis and final prediction model for two classification tasks:

- Classification of user activities (reading, gazing an image and drawing/problem solving)

- Classification of assignments as easy, medium or hard

5.1 Classification of user activities

We were interested in three activities which are now quite common for a lot of tutoring systems:

1. Reading

2. Gazing at an image

3. Drawing/Problem Solving.

The task of reading text is common to almost tutoring applications these days. Gazing images is also very common, and just like reading it's a gaze based information extraction process. The third activity 'drawing' does not just simply refers to the act of drawing an image but also includes the occasional typing and 'problem solving' by the users when they interact with the image which they have already drawn inside the drawing panel.

Let us first discuss the features which can be used to classify the three activities: reading, gazing at an image and drawing/problem solving. We discuss here six features and look into possible explanations for their behavior inside each panel.

1. **Mean Path Distance**

   Mean path distance is defined as the average distance between two consecutive fixations. The scatter-plot shown in Figure 5.1 compares the mean path distance of the users inside the three different panels.
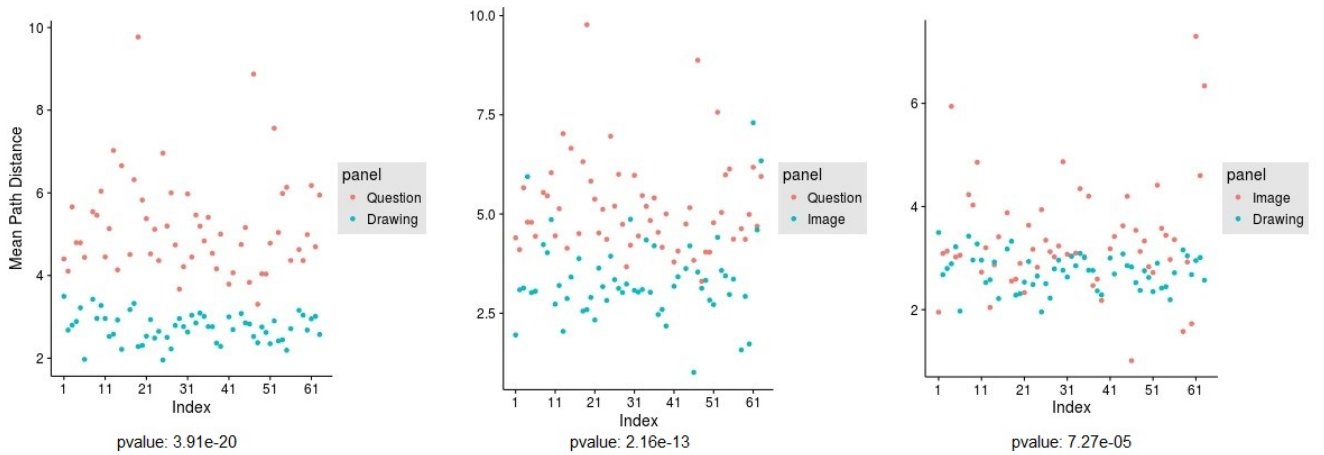

Figure 5.1: Mean path distance

   Mean path distance is relatively higher when reading as compared to the other two activities. While reading, the user knows 'where to move next'. The gaze

movement is quite predictable and hence fast. The eye tracker records the gaze points with a constant frequency of about 30 Hz (30 times every second). If the gaze movement is fast the distance covered in one second increases and so does the average distance between each recorded point. Eye-movement is also fast inside the drawing panel. When a user draws a stroke, the eyes move fast between the two points. Why then, is the Mean Path Distance(MPD) inside the drawing panel so less compared to the question panel? A possible explanation for this can be given by defining a term: s-barrier and look at how it may effect the features related to fixations and saccades.

5.1.1.1   s-barrier

We know that the average distance between each recorded point increases when eyes move fast. These recorded points are fixations when the gaze speed is below a certain threshold. When this speed barrier is crossed, the eye-tracker labels each recorded point as a saccade. We call this the **s-barrier**. The closer we are to the s-barrier from the approaching side, the higher is the Mean Path Distance(MPD). Mean path distance is the average distance between two consecutive fixations. When we cross this speed barrier and move away from it the Mean Saccade Length(MSL) starts increasing and there is no affect on MPD because no more fixations are now recorded. The Figure 5.2 shows the effect of s-barrier on these two features.
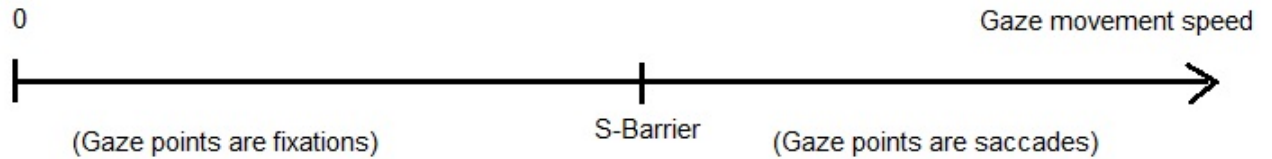
Figure 5.2: Effect of s-barrier on MPD and MSL

The gaze movement inside the question panel is also fast. But, it is not so fast that the eye-tracker only records saccade points. In other words 'the gaze movement speed' is just fast enough to be below the s-barrier'. The mean path distance for reading is thus higher in comparison to the other two activities. In the drawing panel the users draw strokes which require fast gaze movements. But, the speed in that case crosses the s-barrier and hence only effects the mean saccade length. The mean path distance is able to differentiate reading from the other two activities very effectively. But this feature is not very effective in differentiating the user activity inside the image panel from that in the drawing panel.

2. **Mean Saccade Length**

   Reading is a task with fast gaze movements, but this speed for most of the part remains below the s-barrier. However, when users go from one word to the other or change sentences we encounter the fast saccade movements. This results in a relatively longer mean saccade length inside the question panel. Mean saccade length inside the drawing panel is also comparable to that in the reading panel. This is because drawing strokes requires fast gaze movements and this speed is on the right side of the s-barrier in Figure 5.2. The first

scatter-plot in Figure 5.3 shows that the MSL for reading panel and drawing panel is comparable.
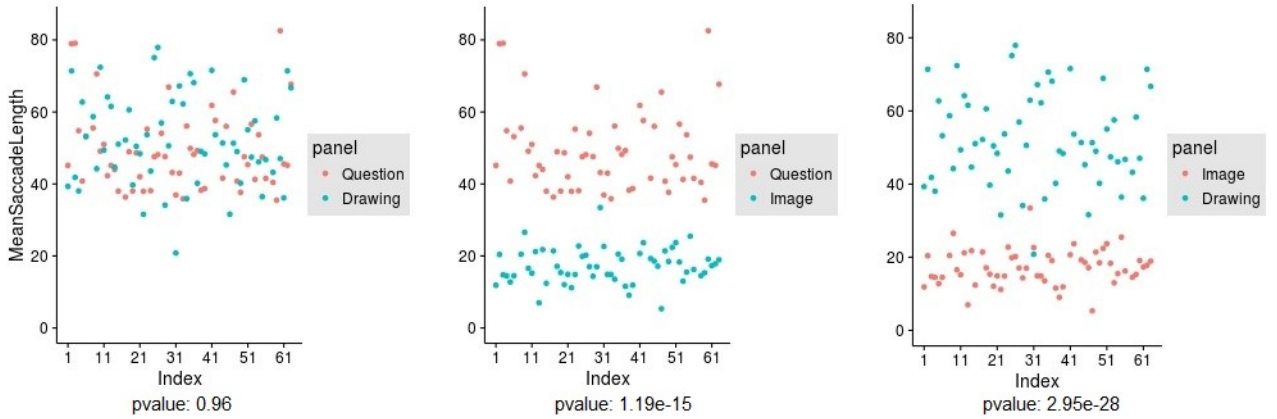
Figure 5.3: Mean Saccade Length (MSL)

Since there are no fast gaze movements in image-gazing the mean saccade length is short for image panel. This makes this feature useful when comparing the image-gazing activity with the other two. This is also reflected in the p-value of each pair which is shown at the bottom of each scatterplot in the Figure 5.3.

3. **FixationSpeedX**

FixationSpeedX refers to the gaze speed along the x-axis when the user goes from one fixation to the other.The scatter-plot shown in Figure 5.4 compares the fixation speed of the user along the x-axis in three different panels.
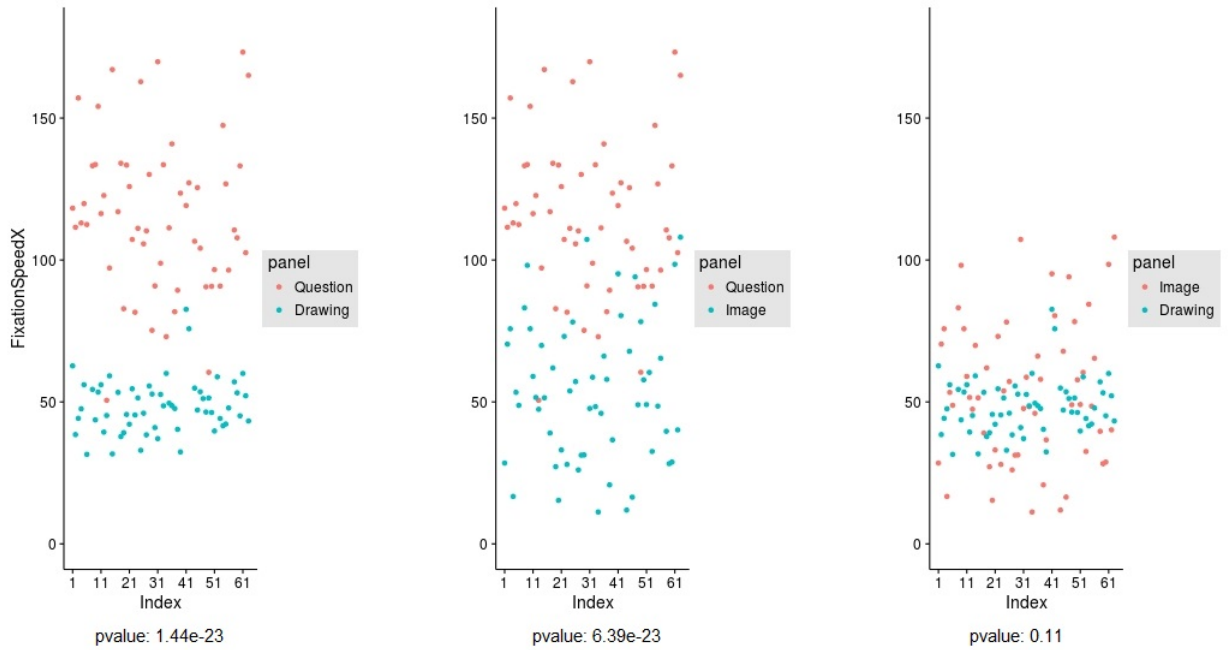
Figure 5.4: Fixation SpeedX

Reading text is an activity in which the user already knows where to look next and hence movement between two consecutive fixations is fast. The pace of horizontal movement is significantly high in reading as compared to both image gazing and drawing. There is fast gaze movement when the users draw strokes inside the drawing panel but the speed of gaze movement is on the right side of the s-barrier. This has no effect on the fixation speed. As can be seen from the scatter-plots and the p-values from the t-tests FixationSpeedX is a very good feature to differentiate reading from the other two activities. However, it is very closely correlated to the Mean path Distance and therefore a redundant feature. We do not use this in our final model.

20

4. **SaccadeSpeedX**

   SaccadeSpeedX is a very important feature. While gazing at the image the occasional inter-saccade movement is at a speed which is very close to the s-barrier hence giving low saccade speed. Reading is quiet fast because of the predetermined gaze movement which also shows up in the inter-saccade movement. In case of drawing panel, most of the saccades are recorded when the strokes are drawn. This is a fast activity and records higher inter-saccade movement. Thus, this feature helps us classify the image-gazing from the other two activities. This can easily be seen in the scatter-plot shown in Figure 5.5
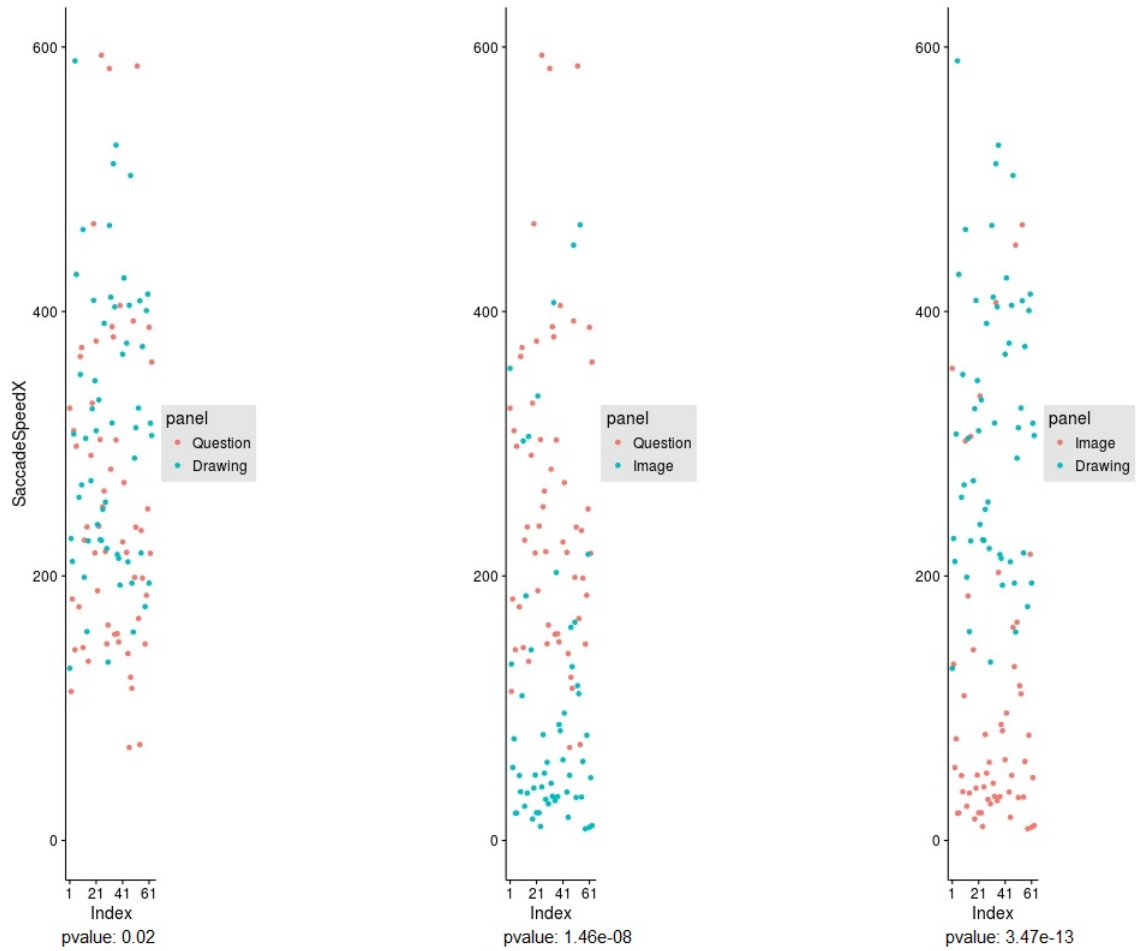
Figure 5.5: Saccade SpeedX

5. **SaccadeSpeedY**

   SaccadeSpeedY behaves no different than the SaccadeSpeedX. We later combine the two features and only calculate Saccade Speed for each activity. The Figure 5.6 shows the scatter-plot for saccade speed along Y-axis.
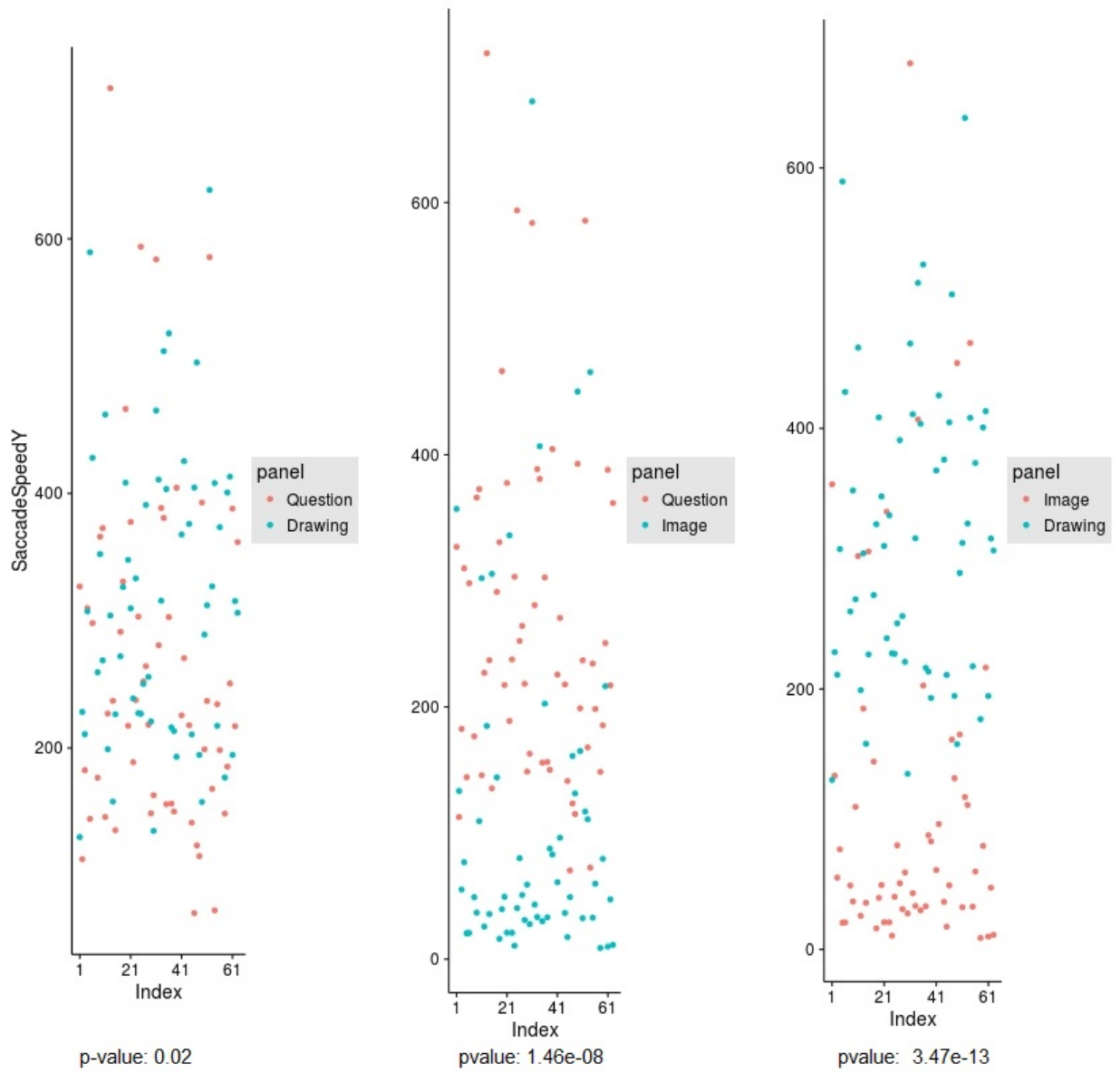
Figure 5.6: Saccade SpeedY

6. **DilationSpeed**

Previous research work has shown that in a problem-solving scenario there is a link between mental work load and pupil diameter. Bailey [3] and Iqbal [19]

use this correlation between pupil dilation and and cognitive load to suggest a framework for detecting task boundaries. Pupil activity has also been shown to be sensitive to goal-directed discrete movements [21]. This gave us the idea to use dilation speed of the pupil as a feature which can help us differentiate the drawing panel activity from the other two. The scatter-plots in Figure 5.7 show that the dilation speed for the drawing panel is significantly higher than the other two. The same is also reflected in the p-values for each of these pairs.
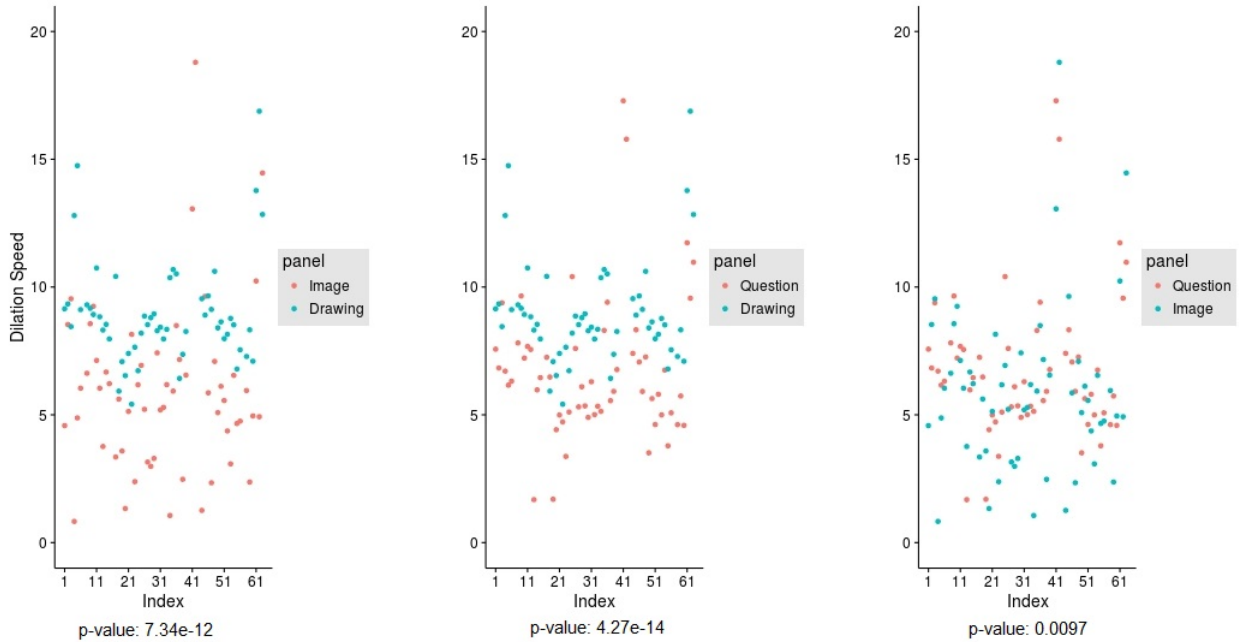
Figure 5.7: Dilation Speed

5.1.2 Principal Component Analysis(PCA)

We applied principal component analysis on the features which we have looked at so far to reduce the dimensionality of our data. This is important because we do

not have a large number of data points. Reducing the dimensionality of the data helps us prevent over-fitting. It also makes it easier to visualize the data. The Figure 5.8 shows a scatter-plot of the data when we use only the top two principal components for projection. This shows that the three classes are sufficiently separated when projected on these components. The Figure 5.9 shows a 3D-scatter-plot of the same data but with the first three components from PCA. We have tilted the axes to better visualize the scatter-plot in 2D plane.
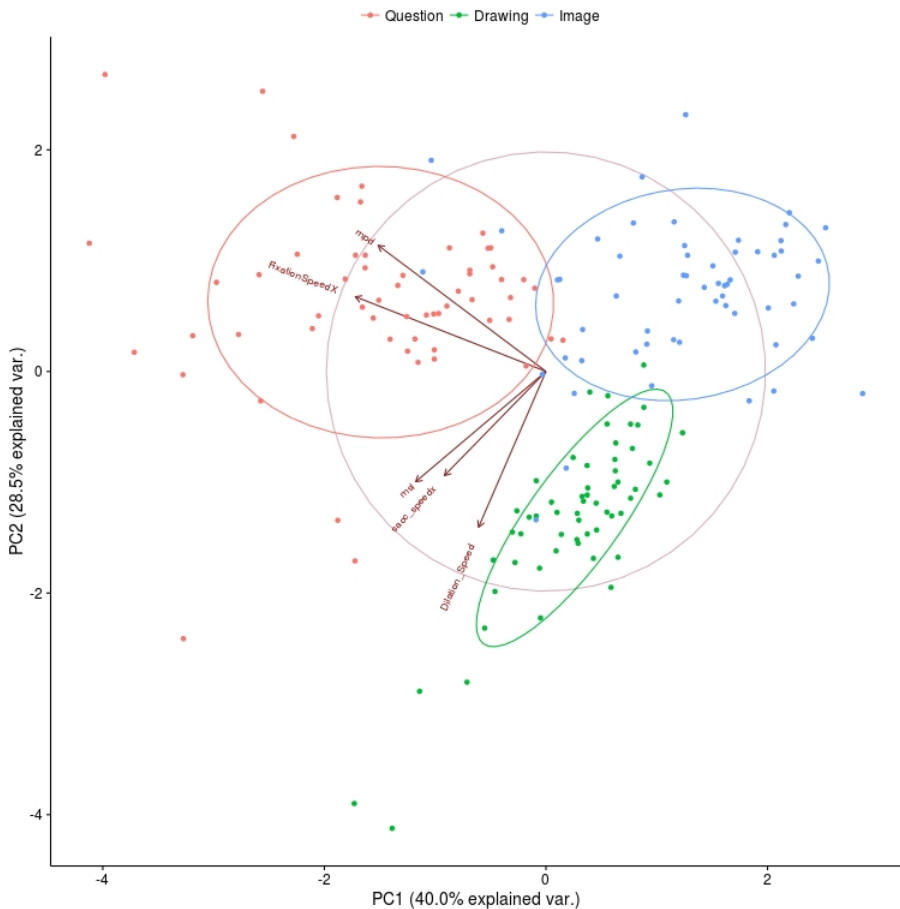
Figure 5.8: Scatter-plot using the top two components from PCA.
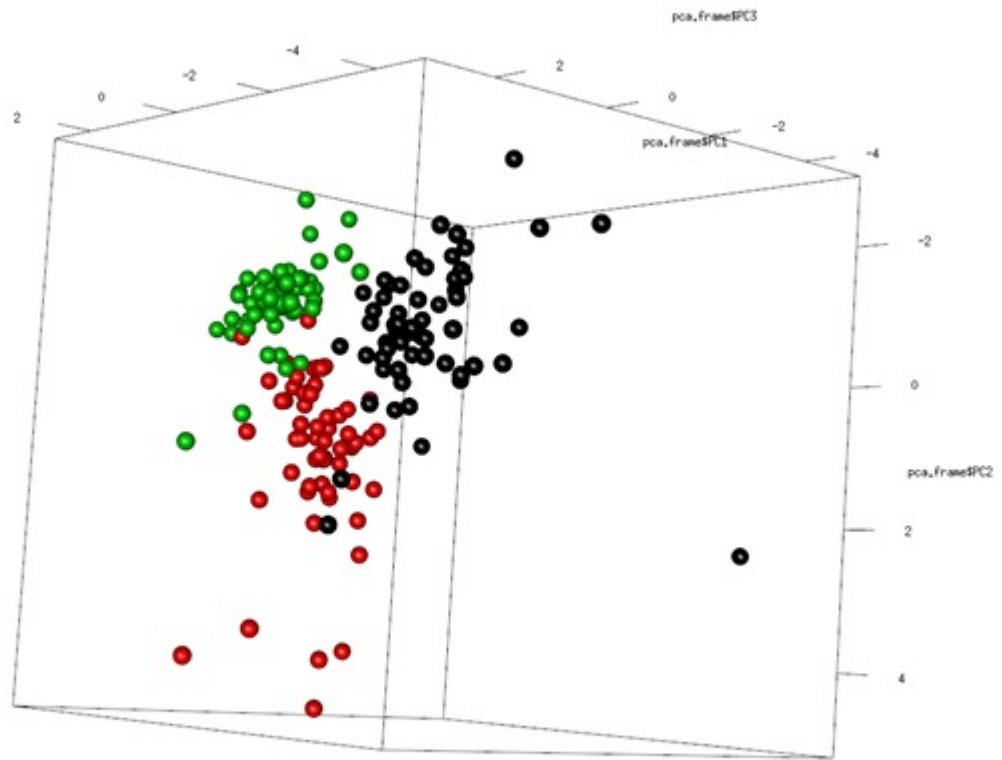
Figure 5.9: 3D scatter-plot using the top three components from PCA.

5.1.3   Feature selection

We have 174 data points for training and testing our model. Six features can cause over-fitting. Therefore, before using this data to train a classifier we reduced the dimensionality of this dataset. This could be achieved by PCA or by variable subset selection. PCA helped us visualize the data, but for building the model we used subset selection because it makes the model simpler. The first step was to look for highly correlated features. Table 5.1 shows the correlation matrix for six features.

	SaccadeSpeedx	MSL	MPD	Dilation_Speed	FixationSpeedX
SaccadeSpeedx	1	0.319	0.119	0.217	0.131
MSL	0.319	1	0.126	0.326	0.284
MPD	0.119	0.126	1	-0.140	0.778
Dilation_Speed	0.217	0.326	-0.140	1	0.137
FixationSpeedX	0.131	0.284	0.778	0.137	1

Table 5.1: Correlation matrix

Following Cohen's [10] conventions to interpret effect size, we can easily say that FixationSpeedX is highly correlated to the Mean Path Distance. This can also be observed by comparing the scatter-plots of these two features (Figure 5.4 and 5.1). We compare the area under the ROC curve to rank the features(importance). Table 5.2 and Figure 5.10 show the variables sorted by importance across the three classes. We also wanted to know how many features do we really want to build an accurate model and avoid over-fitting.

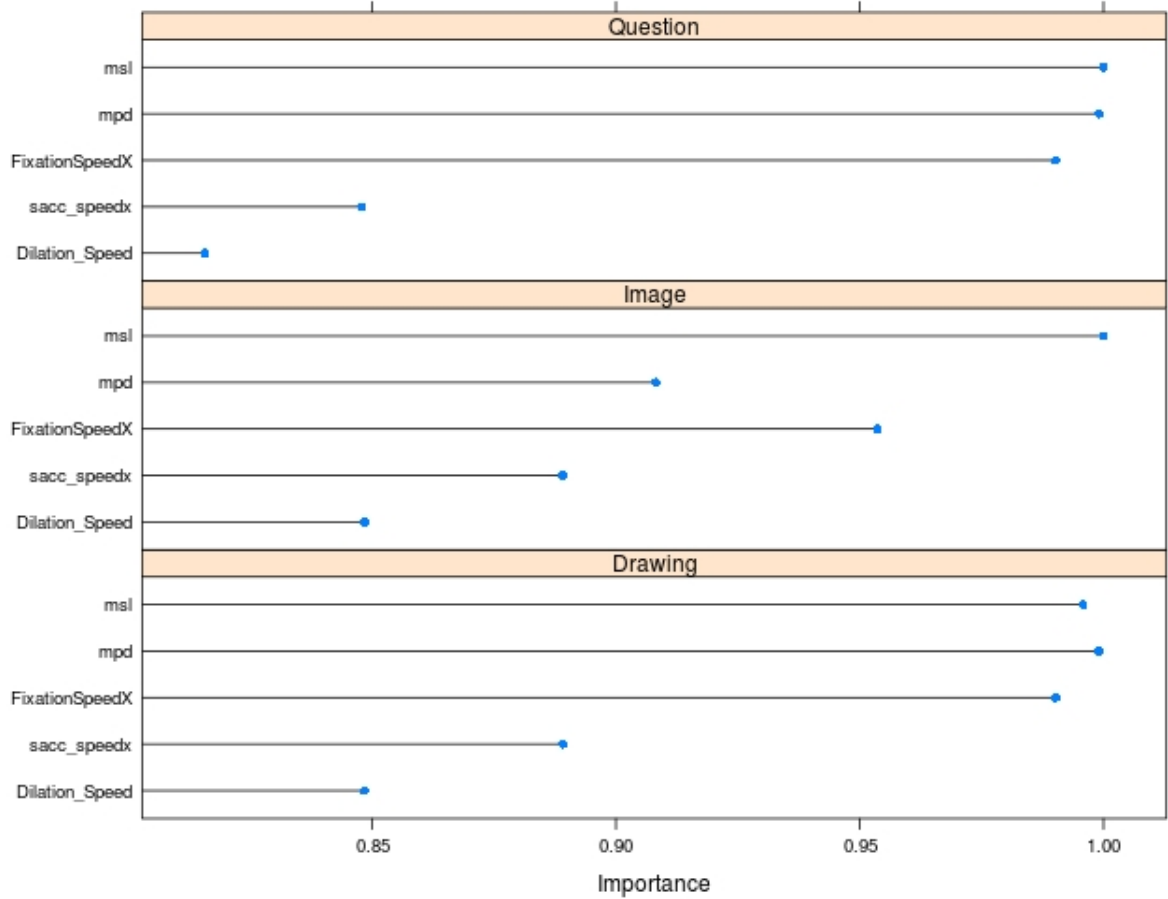|	Question	Drawing	Image
MSL	1.0	0.9958	1
MPD	0.9991	0.9991	0.9081
FixationSpeedX	0.9902	0.9902	0.9536
SaccadeSpeedx	0.8478	0.8891	0.8891
Dilation_Speed	0.8157	0.8484	0.8484

Table 5.2: Variable importance

Figure 5.10: Importance of features

We use the Reverse feature elimination method to find out the best features to use. From the plot shown in Figure 5.11 we can see that by using only the top two features: Mean Saccade length (MSL) and Mean Path distance(MPD) we get fairly high accuracy.
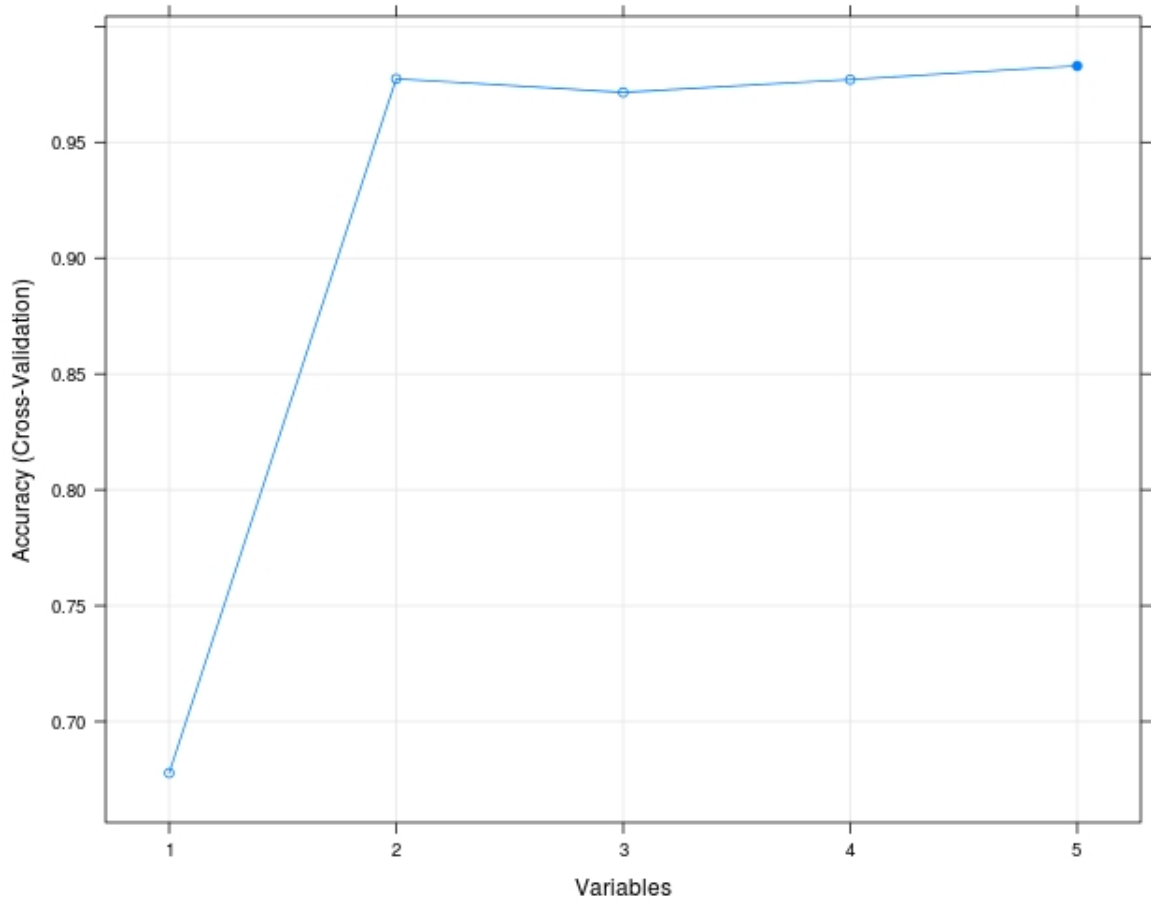
Figure 5.11: Accuracy with features

5.1.4   Classification results

Classification of activities is a multi-class problem. This makes the multi-class classifier in Weka a very good candidate for choice of classifier. Baseline performance for comparison was established using a majority classifier. There are equal number(58) of data points for all three classes. Therefore the majority classifier gives an accuracy of 33.33%

A multiclass classifier gives an accuracy of 96.55% and f-measure of 0.966 when tested with 10-fold cross validation. We have tested other classifiers as well with the same data and Table 5.3 shows the results for those models.

Classifier	Accuracy percentage
Multi-class Classifier	96.55
NaiveBayes	95.977
J48	96.55
Random Forest	97.12

Table 5.3: Classifers

The RandomForest classifier with maxDepth 0 gives the best result with an accuracy of 97.12% at an f-measure of 0.97. Table 5.4 shows the confusion matrix for this classification.

Classified as ->	Question	Drawing	Image
Question	57	1	0
Drawing	2	55	1
Image	1	0	57

Table 5.4: Confusion matrix - randomforest classifier

5.2   Classification of assignments as easy, medium and hard

In this section we present the results for the prediction of difficulty level (easy, medium or hard) of tasks (assignments in this case). We first discuss the process of feature selection. Then, we describe the behaviour of most important features which were used in this model. Finally, we give results of the various classifiers which were tested.

5.2.1   Dropping redundant features

For the difficulty level prediction problem, we start by analysing twenty one features calculated from gaze data. First step was to get rid of the redundant features. We again employed the Cohen's criteria of choosing the highly correlated features and dropped those. Table 5.5 shows a list of the original features with the highly correlated features highlighted in color.These redundant variables were dropped at this stage and not used for further analysis. Figure 5.12 shows the most important features for each of the three classes. These features are ranked by area under the ROC curve. The area under the ROC curve of a variable w.r.t a class gives an estimate of how well the variable can differentiate this class from the other two classes. This only gives an approximate idea about the usefulness of the feature if this was used as a lone predictor to train a classifier.

Features
FixationSpeedX_qpanel
FixationSpeedX_dpanel
FixationSpeedX_ipanel
MPD_qpanel
MPD_dpanel
MPD_ipanel
Dilation_Speed_qpanel
Dilation_Speed_dpanel
Dilation_Speed_ipanel
Range_PupilSize_qpanel
Range_PupilSize_dpanel
Range_PupilSize_ipanel
msl_qpanel
msl_dpanel
msl_ipanel
s_speedx_qpanel
s_speedx_dpanel
s_speedx_ipanel
s_speedy_qpanel
s_speedy_dpanel
s_speedy_ipanel

Table 5.5: List of features. Highlighted features are highly correlated
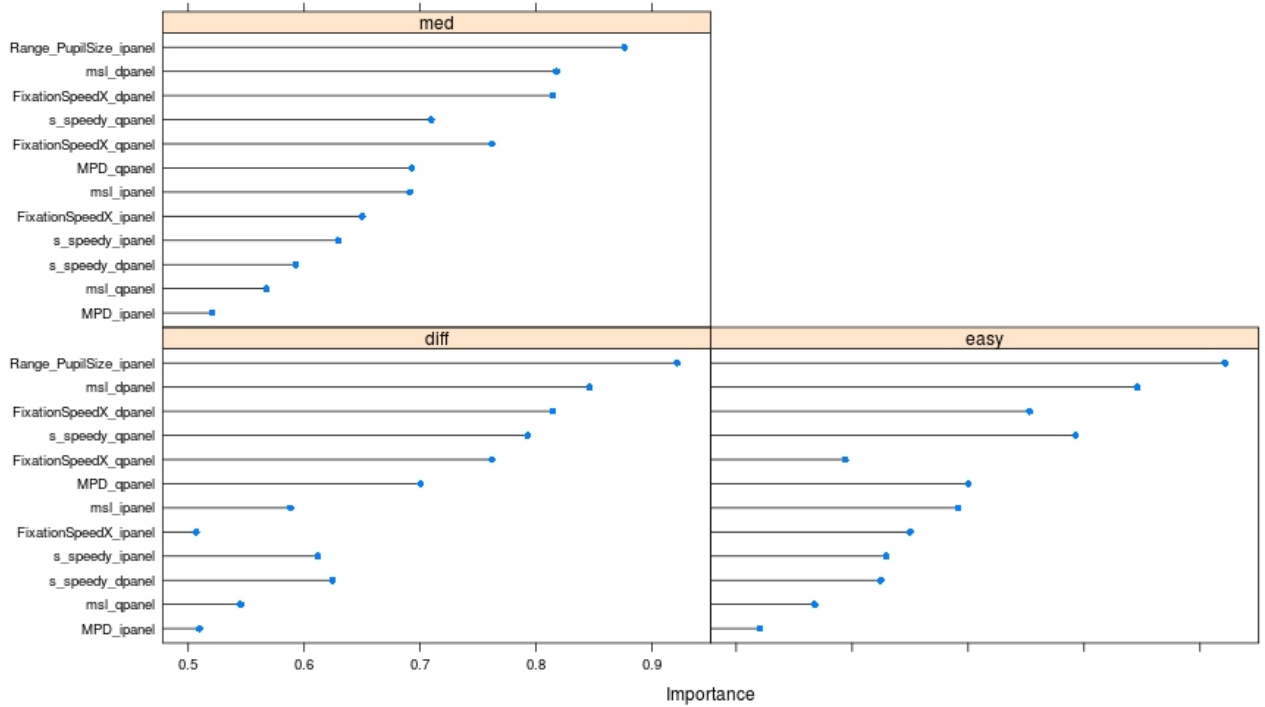
Figure 5.12: Importance of features ranked by area under ROC curve

However, this is not the case and we intend to to find the best subset of features which can be used together to build a model. For this we perform a recursive process of feature elimination.

5.2.2 Recursive feature elimination

Next, we use Recursive Feature Elimination (RFE) with RandomForest model to get the best suited subset of features. It tells us the information that we can get an optimum prediction performance only when we use the best seven features. The Figure 5.13 shows that if we use seven features the accuracy can go over 80% and is an optimum choice. These features are shown in Table 5.6.
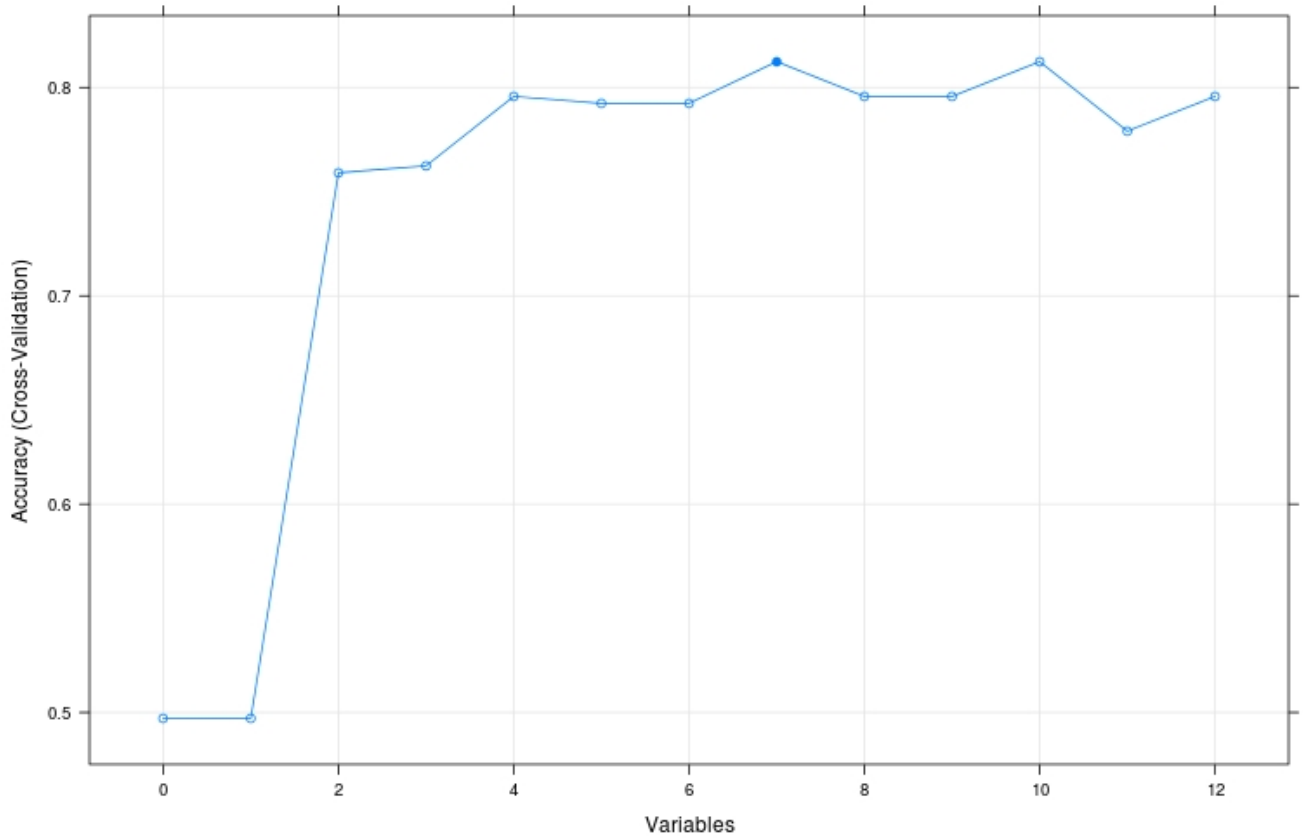
Figure 5.13: Choosing the optimum number of features

5.2.3   Principal Component Analysis(PCA)

We do not have a large data set to train and hence there is a danger of over-fitting the data. We therefore turned to PCA to reduce the dimensionality of the data further to have no more than three final components to train the classifier. The Figure 5.14 shows a scatter-plot when the top two components of the PCA result were used to plot the three classes.

Best Features
msl_dpanel
Range_PupilSize_ipanel
s_speedy_qpanel
FixationSpeedX_dpanel
FixationSpeedX_qpanel
MPD_qpanel
msl_ipanel

Table 5.6: List of best features.

The scatter-plot shows that the classes difficult and easy have a slight overlap but, the class medium overlaps both easy and difficult hence making it difficult to predict.
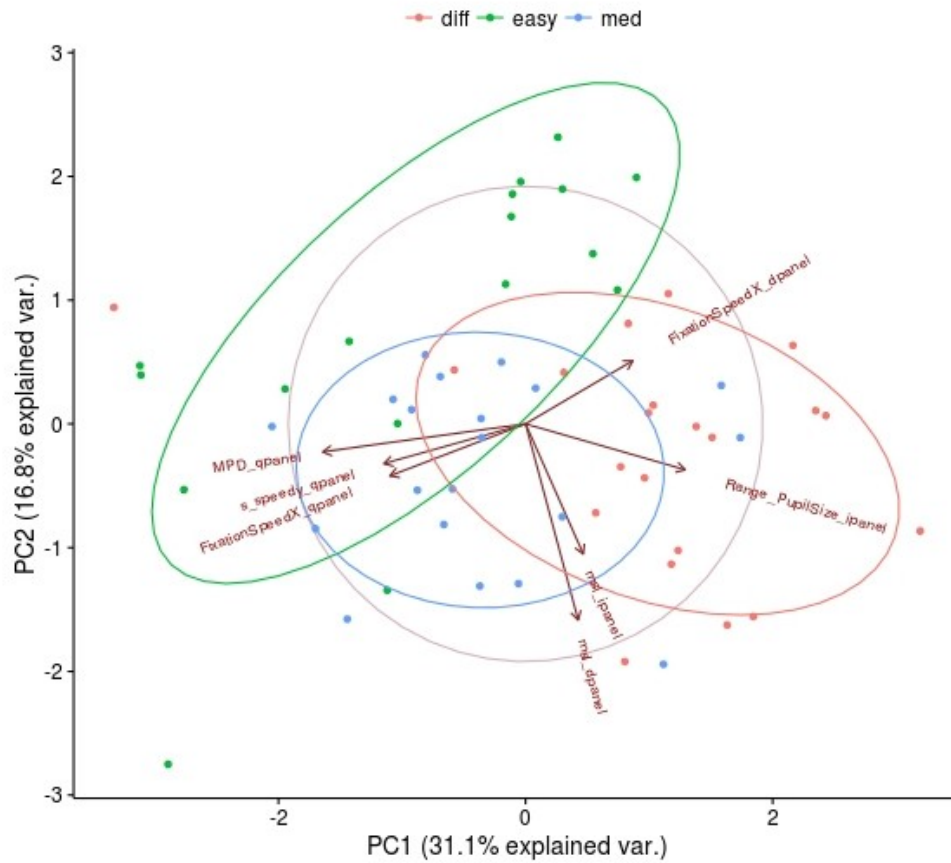
Figure 5.14: Principal Component Analysis (PCA)

5.2.4 Selected features

Following are the chosen features:

1. **Mean Saccade Length in Drawing panel**

   Mean saccade length inside drawing panel is the most important feature in this analysis. When the assignments get harder, the problem figures get more complicated with more and often longer strokes. This leads to longer saccadic movements in which the mean distance between saccades is high. The Figure 5.15 shows the diagrams from an easy problem and a difficult/complicated

problem. The longest stroke in the square is it's side where as in the diagram on right there can be several strokes (depending on the way user draws this picture) which are longer than that. This is probably is a reason for the more difficult problems having longer inter-saccadic distances.
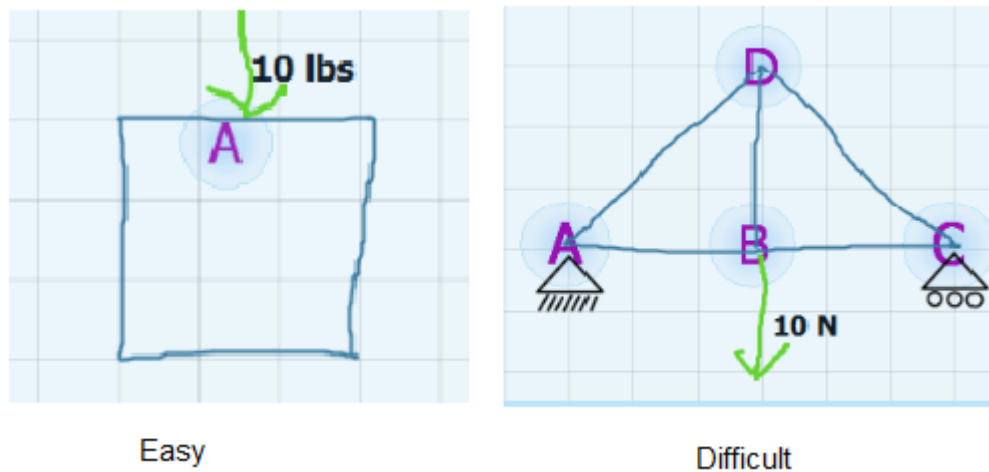
Figure 5.15: Problem diagrams from easy(on left) and difficult(on right) problems

2. **Saccade Speed in X-direction while reading**

   As we go from easier problems to harder problems the users tend to read slow because they take more time to comprehend the text. This leads to a significant difference in their inter-saccadic gaze speed. The speed is highest for easier problems and decreases as we move towards more difficult problems.

3. **Range of Pupil Size in i_panel**

   The pupil activity inside the image panel(i_panel) significantly changes as the user starts looking at more complicated diagrams from the more difficult problems. The range is narrower for easier problems and increases as the diagrams

get more complicated.

4. **Pupil Dilation Speed in i_panel**

   The dilation speed of the pupil inside the image panel is also an important feature which varies from easier to more difficult problems.

   The Table 5.7 shows the p-values for these features which we get from the regression analysis. The R-square value for this is .67 i.e. the features explain 67% of the data.

Feature	p-Value
MeanSaccadeLength_dpanel	4.5782e-06
SaccadeSpeedX_qpanel	1.30682e-05
RangePupilSize_ipanel	1.59684e-05
PupilDilationSpeed_ipanel_ipanel	2.215e-02

Table 5.7: Features and their p-values in regression analysis

5.2.5   Classification results

We use the majority classifier in Weka: ZeroR to establish the baseline classification performance. We use the top three components from the PCA to train the classifiers. The LibSVM classifier in Weka gives the best accuracy of 81.03% and an f-measure of .81. Table 5.9 shows the confusion matrix for this classification model. We definitely need more data to build a better and accurate model. The Table 5.8 gives the result from other classifiers which were also tested.

An f-measure of 0.81 was reported for classification using 10-fold cross validation. The Table 5.9 shows the confusion matrix for this model.

Classifier	Accuracy Percentage
ZeroR	36.21
Multiclass Classifier	77.58
LibSVM	81.03
NaiveBayes	79.31
RandomForest	68.96

Table 5.8: Accuracy of classification

Classified as->	easy	diff	med
easy	13	1	3
diff	1	18	2
med	2	2	16

Table 5.9: Confusion matrix - LibSVM using 10-fold cross validation

6. CONCLUSION

In this thesis, we have tackled two classification problems. The first one is how accurately can we classify the activity of a user as reading, gazing-image or drawing in a problem-solving scenario and the second was the ability to predict the task difficulty if the first problem of activity classification is solved with hundred percent accuracy. We designed an experiment using Mechanix in which the users attempt varied levels of problems on an interface which engages the user to do all three kinds of activities. We rely on an eye-tracker to give us gaze data of the user.

In our approach towards analysing gaze data, we start by cleaning and pre-processing the data. We calculate 'pure' gaze based features and conduct an exploratory analysis for each of these to estimate their usefulness in a learning model. For the activity classification problem, we find six useful features. Reverse Feature Estimation (RFE) based on RandomForest showed that Mean Saccade Length(MSL) and Mean path Distance(MPD) only can give up to 97% accuracy in predicting the activity. For the second problem, we first get rid of the redundant (highly correlated) features to narrow down the options and then apply the RFE to get a subset of best seven features only. Because of the constraint on the amount of data we apply PCA to get the best three principal components which are used to train a LibSVM classifier. This model was able to predict the difficulty level of the problem with an accuracy of about 81%.

Overall, we showed that gaze-based features can be used to infer the user-activity on an interface and estimate the difficulty level of a problem that engages the user in the three common activities: reading, gazing at an image and drawing.

7. FUTURE WORK

The future steps of this work include a prediction model which combines both these results to predict the task difficulty without knowing the activity in advance. Another step in the same direction would be to test such a model by developing a real-time system that dynamically captures and classifies user activities based on gaze movement. This system should be then tested for platforms which allow hand-written/drawn text and images such as the Microsoft Hub and the plain old paper as well. Work will also be required to increase the accuracy of our classifier which predicts the task difficulty. More training data is definitely one of the limiting factors. Therefore, we would like to do some more users studies and see if it improves the accuracy of the model.

Findings from this experiment can also be used in other applications. We have learnt about the behaviour of several important features through this experiment. We would like to see if these features can be used in other scenarios like the patient diagnosis for diseases which affect human cognitive powers (examples: Alzheimer's disease, dyslexia etc).

REFERENCES

[1] Olufunmilola Atilola, Martin Field, Erin McTigue, Tracy Hammond, and Julie Linsey. Evaluation of a natural sketch interface for truss fbds and analysis. In *Frontiers in Education Conference (FIE), 2011*, pages S2E–1. IEEE, 2011.

[2] Olufunmilola Atilola, Stephanie Valentine, Hong-Hoe Kim, David Turner, Erin McTigue, Tracy Hammond, and Julie Linsey. Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 28:169–192, 5 2014.

[3] Brian P. Bailey and Shamsi T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interactions*, 14(4):21:1–21:28, January 2008.

[4] R. Bednarik, T. Gowases, and M. Tukiainen. Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research*, 3(1):110.

[5] Roman Bednarik, Hana Vrzakova, and Michal Hradis. What do you want to do next: A novel approach for intent prediction in gaze-based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 83–90, New York, NY, USA, 2012. ACM.

[6] Ralf Biedert, Georg Buscher, Sven Schwarz, Manuel Möller, Andreas Dengel, and Thomas Lottermann. The text 2.0 framework: Writing web-based gaze-

controlled realtime applications quickly and easily. In *Proceedings of the 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction*, EGIHMI '10, pages 114–117, New York, NY, USA, 2010. ACM.

[7] Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 123–130, New York, NY, USA, 2012. ACM.

[8] Andreas Bulling, Jamie A. Ward, Hans Gellersen, and Gerhard Tröster. Robust recognition of reading activity in transit using wearable electrooculography. In *Proceedings of the 6th International Conference on Pervasive Computing*, Pervasive '08, pages 19–37, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] Siyuan Chen, Julien Epps, and Fang Chen. Automatic and continuous user task analysis via eye activity. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 57–66, New York, NY, USA, 2013. ACM.

[10] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic Press, New York, 1977.

[11] Danielle Cummmings, Francisco Vides, and Tracy Hammond. I don't believe my eyes!: geometric sketch recognition for a computer art tutorial. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 97–106. Eurographics Association, 2012.

[12] O. Dimigen, W. Sommer, A. Hohlfeld, A. M. Jacobs, and R. Kliegl. Coregistration of eye movements and eeg in natural reading: analyses and review. *Journal of Experimental Psychology:General*, 2011.

[13] Daniel Dixon, Manoj Prasad, and Tracy Hammond. icandraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2010.

[14] Geoffrey B. Duggan and Stephen J. Payne. Skim reading by satisficing: Evidence from eye tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1141–1150, New York, NY, USA, 2011. ACM.

[15] Myse Elmadani, Antonija Mitrovic, Amali Weerasinghe, and Kourosh Neshatian. Investigating student interactions with tutorial dialogues in eer-tutor. *Research and Practice in Technology Enhanced Learning*, 10(1), 2015.

[16] Martin Field, Stephanie Valentine, Julie Linsey, and Tracy Hammond. Sketch recognition algorithms for comparing complex and unpredictable shapes. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2436–2441. AAAI Press, 2011.

[17] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[18] Tracy Tracy Hammond Hammond. Enabling instructors to develop sketch recognition applications for the classroom. In *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, pages S3J–11. IEEE, 2007.

[19] Shamsi T. Iqbal, Xianjun Sam Zheng, and Brian P. Bailey. Task-evoked pupillary response to mental workload in human-computer interaction. In *CHI '04*

Extended Abstracts on Human Factors in Computing Systems, CHI EA '04, pages 1477–1480, New York, NY, USA, 2004. ACM.

[20] Robert J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transaction on Information Systems*, 9(2):152–169, April 1991.

[21] Xianta Jiang, M. Stella Atkins, Geoffrey Tien, Roman Bednarik, and Bin Zheng. Pupil responses during discrete goal-directed movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2075–2084, New York, NY, USA, 2014. ACM.

[22] Carpenter Patricia A. Just, Marcel A. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87(4):329–354, 1980.

[23] Kourtney Kebodeaux, Martin Field, and Tracy Hammond. Defining precise measurements with sketched annotations. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 79–86. ACM, 2011.

[24] H Kim, P Taele, S Valentine, J Liew, and T Hammond. Developing intelligent sketch-based applications to support children's self-regulation and school readiness. In *2014 Intelligent User Interfaces Workshop on Sketch Recognition*, pages 1–8. ACM, 2014.

[25] H Kim, S Valentine, P Taele, and T Hammond. Easysketch: A sketch-based educational interface to support childrens self-regulation and school readiness. In *The Impact of Pen and Touch Technology on Education*, pages 35–46. Springer, 2015.

[26] Hong-hoe Kim, Paul Taele, Stephanie Valentine, Erin McTigue, and Tracy Hammond. Kimchi: a sketch-based developmental skill classifier to enhance pen-driven educational interfaces for children. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pages 33–42. ACM, 2013.

[27] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(1):1–26, 2008.

[28] Kai Kunze, Hitoshi Kawaichi, Kazuyo Yoshimura, and Koichi Kise. Towards inferring language expertise using eye tracking. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 217–222. ACM, 2013.

[29] Tomasz D. Loboda, Peter Brusilovsky, and Jöerg Brunstein. Inferring word relevance from eye-movements of readers. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 175–184, New York, NY, USA, 2011. ACM.

[30] Pascual Martínez-Gómez and Akiko Aizawa. Recognition of understanding level and language skill using measurements of reading behavior. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 95–104, New York, NY, USA, 2014. ACM.

[31] Vidhya Navalpakkam, Justin Rao, and Malcolm Slaney. Using gaze patterns to study and predict reading struggles due to distraction. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1705–1710, New York, NY, USA, 2011. ACM.

[32] Trevor Nelligan, Seth Polsley, Jaideep Ray, Michael Helms, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based educational interface. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, IUI Companion '15, pages 53–56, New York, NY, USA, 2015. ACM.

[33] Joshua M Peschel and Tracy Anne Hammond. Strat: a sketched-truss recognition and analysis tool. In *2008 International Workshop on Visual Languages and Computing (VLC) at the 14th International Conference on distributed Multimedia Systems (DMS)*. Knowledge Systems Instistute, 2008.

[34] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, pages 618–660, 1998.

[35] Jaana Simola, Jarkko Salojärvi, and Ilpo Kojo. Using hidden markov model to uncover processing states from eye movements in information search tasks. *Cogn. Syst. Res.*, 9(4):237–251, October 2008.

[36] Paul Taele and Tracy Hammond. A geometric-based sketch recognition approach for handwritten mandarin phonetic symbols i. In *2008 International Workshop on Visual Languages and Computing (VLC) at the 14th International Conference on distributed Multimedia Systems (DMS)*, 2008.

[37] Paul Taele and Tracy Hammond. Using a geometric-based sketch recognition approach to sketch chinese radicals. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*, pages 1832–1833. AAAI Press, 2008.

[38] Paul Taele and Tracy Hammond. Hashigo: A next-generation sketch interactive system for japanese kanji. In *Twenty-First IAAI Conference*, 2009.

[39] Paul Taele and Tracy Hammond. Lamps: A sketch recognition-based teaching tool for mandarin phonetic symbols i. *Journal of Visual Languages and Computing*, 21(2):109–120, 2010.

[40] Paul Taele and Tracy Hammond. Enhancing instruction of written east asian languages with sketch recognition-based intelligent language workbook inter-

faces. In *The Impact of Pen and Touch Technology on Education*, pages 119–126. Springer, 2015.

[41] Paul Taele, Joshua Peschel, and Tracy Hammond. A sketch interactive approach to computer-assisted biology instruction. In *Proc. IUI 2009 Workshop on Sketch Recognition*, 2009.

[42] Stephanie Valentine and Martin Field. A shape comparison technique for use in sketch-based tutoring systems. In *Proceedings of the 2011 Intelligent User Interfaces Workshop on Sketch Recognition (Palo Alto, CA, USA, 2011), IUI*, 2011.

[43] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based tutoring system for statics courses. In *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference (IAAI)*, 2012.

[44] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong-hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. Mechanix: a sketch-based tutoring and grading system for free-body diagrams. *AI Magazine*, 34(1):55, 2012.

[45] F Vides, P Taele, H Kim, J Ho, and T Hammond. Intelligent feedback for kids using sketch recognition. In *ACM SIGCHI 2012 Conference on Human Factors in Computing Systems Workshop on Educational Interfaces, Software, and Technology.* ACM, 2012.

[46] Songhua Xu, Hao Jiang, and Francis C.M. Lau. User-oriented document summarization through vision-based eye-tracking. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 7–16, New York, NY, USA, 2009. ACM.

[47] Alfred L. Yarbus. *Eye movements and vision.* Springer, 1967.

ROC CURVE

In this study we have used area under the ROC curve(AUROCC) to judge individual features for their individual classification capability. We know that a good diagnostic test is one with high sensitivity (high true positive rate) and specificity (high true negative rate). Also, specificity decreases as the sensitivity increases. Therefore, we have to choose a test which gives us the best balance between both. The ROC curve which is a curve between sensitivity and 1-specificity helps us choose between features. The Figure A.1 shows ROC curves for a good diagnostic test and an uninformative test.
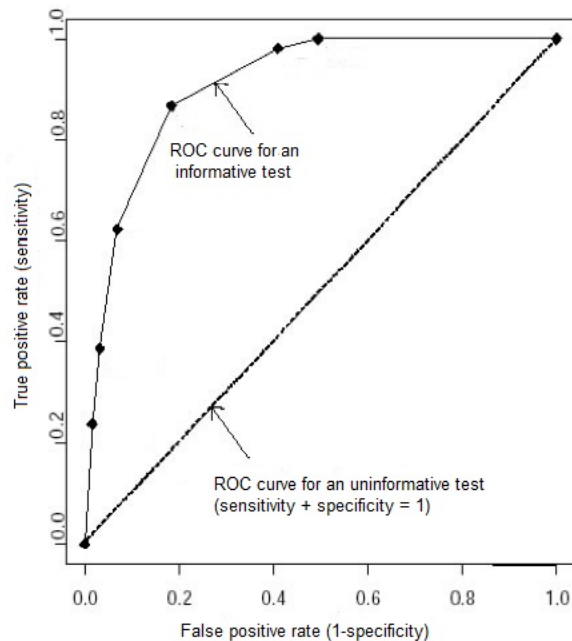

Figure A.1: ROC curves

The area under the ROC curve (AUROCC) is a reflection of how good the test is at discriminating between two classes.The area under an ROC curve may vary from 0.5 to 1. Higher the AUROCC, the better the test (in this case feature). Therefore, we test each feature in our study by calculating the AUROCC for that feature and then compare it with others to rank the features.

APPENDIX B

PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis is a powerful tool for analysing data. It is a way of identifying patterns in data. If the data has high dimensionality, it often becomes hard to find out the patterns in data. PCA can be used to reduce this dimensionality and make the graphical representation of data easier. Sometimes, we also have to reduce the dimensionality of the data to avoid over-fitting. We can use PCA to reduce the dimensionality there as well.