Dissertation - Detecting User Engagement Using Mouse Tracking Data

David Saunders (910995)

September 2020

Abstract

Write abstract here

Contents

1 Motivation

Detecting use engagement is an important science across multiple disciplines.

Find a study that says during any task / any crowdsourced task X% of people are not paying attention. This could jeopardise the results from any online if we cannot be sure users are actually paying attention to the task they are being paid to complete.

It is a hard problem because, among other things I will touch on, it is hard to quantify exactly what user engagement is exactly. One author defined it as 'XYZ' [look at my previous work for references].

However the previous authors have failed to detect and measure user engagement as X,Y,Z.

1.1 Contributions

In this project my contributions to fix it are, -a system to classify users, a way of visualising their mouse paths, ways to directly and quantitatively compare different users, and a multistage semi-supervised based binary classification output to answer the question of 'are users engaged'.

2 Related Work

All related work about actual other attempts to detect user engagement from mouse tracking data. Other subsections will be like miniature literature reviews or something?

2.1 Semi Supervised Learning

This will be a key aspect of the project as we only have definitive labels for part of our data. This reflects the challenges of real world data, by some estimates only 2% (made up number) of all data is structured and labelled, the rest is unstructured [find reference].

In a sense the data here is labelled. All day belongs to 2 classes, online turk user, or in person lab study user. However on the other hand the data is not labelled for the task I would like to explore. The goal of this project is to identity which users were paying attention. We have assumed that we can infer that lab study users will be paying attention, so we can say that those samples are labelled. However the rest of the samples we have which are all of the online data are unlabelled. Therefore this is a kind of semi supervised learning problem where we only have a small percentage of our samples labelled, and only confident labels for one class. (See if a paper on this exists, semi supervised binary classification with only labels for one class.)

2.2 N-Grams

This paper has a nice scientific explanation of n-grams [1]. Either cite this paper or more likely look at their reference for n-grams and cite that.

2.3 Hidden Markov Models

Paper Tom send me about HMM for text classification that I might be able to use [2].

Paper claims to use HMM for spam detection, I think they actually just use it to detect misspellings of words or something which is used by spam to hide from filters [3]. Probably could find a better spam detecting HMM, I just like how this almost does something different from the title, which my diss will end up doing.

This paper was recommended by someone on stack overflow as an old influential paper in the field with tens of thousands of references [4]. Called a tutorial on HMM and its so old so original source. Would definitely be good to reference if I include any of the mathematics behind HMMs.

Someone's dissertation on the topic of generating synthetic data with HMMs. This can be a good way to create synthetic data [5].

Maybe this would belong under implementation?

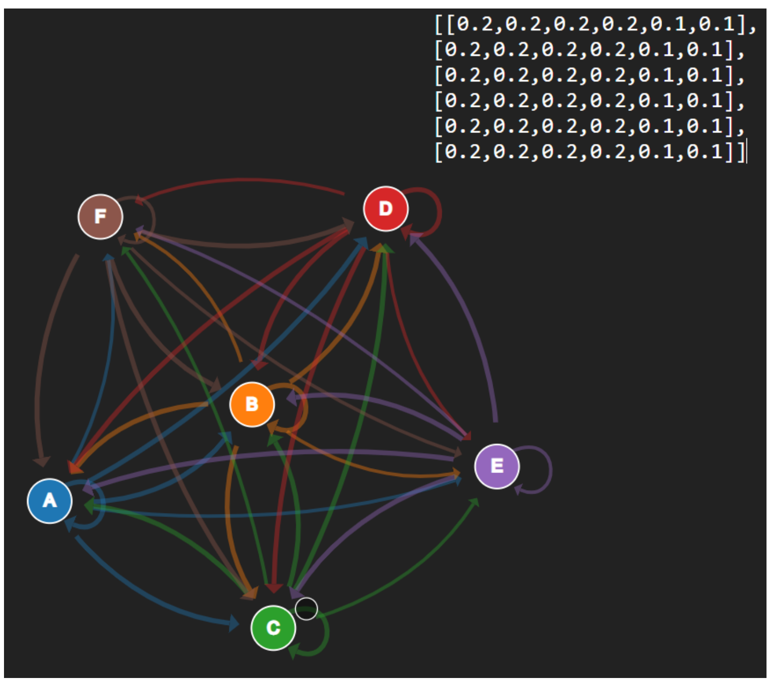

Figure 1: Diagram of a Markov Chain model representing a users mouse position

Figure 1 shows the states of a markov model of my system with states A-E representing sliders 1-5 and state F representing an html element. In the top right we can see a possible transition matrix of our system.

3 Implementation

4 Methodology

The assumption of this project is that labs are paying attention and turks are not. Therefore if we get any outliers from the turk data, but that are similar to the lab data, then we will say that that online turk user was paying more attention than his peers, and that they were paying attention.

This study (ref) says that only 10% of all people / turk users pay attention during a task. We will look at the 10% (30ish) of the turk data that looks like it is lab data and day that they were paying attention. This is just the assumption we have made for the project, unfortunately the dataset isn't extensive enough for us to fully test this hypothesis.

5 Data Pipeline

When planning and completing this project many decisions were made about the steps taken to convert the raw data to a finished product/classification. This section may act as an overview of the project, detailing the different sections of work, what they may contain, and the order in which they will be completed.

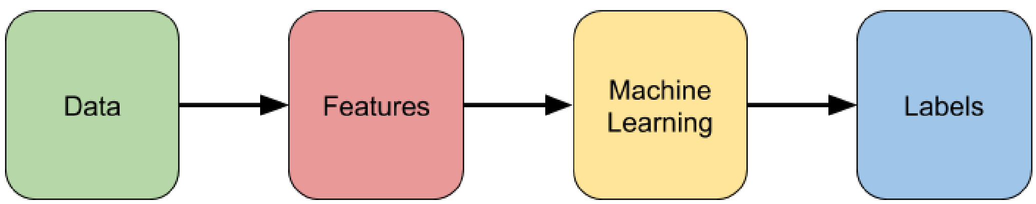

Figure 2: Diagram of the Data Pipeline of the project.

5.1 Data

Here I will summarise what I have done to the data. As previously mentioned the data was gathered through a lab study and an online study. The purpose of the online study was to gather a larger amount of data that was possible to do so in person. The aim of this was to make any results more statically significant. Data was recorded in a big JSON dump with lots of irrelevant and repealed data relating to the users background and not their mouse movements. Pythons JSON couldn't directly convert the data as mouse events were stored as a nested JSON dictionary and there were errors in the way it was written making it invalid JSON. Took ages and should explain more indepth, but then finally got data into a tabular data format of a csv which I am more comfortable working with. Ended up with over 100,000 lines? Might have been more.

This leads to the problem of imbalanced data samples. There were approximately 11 lab data and 400 online data, meaning that there were 40x as many data samples from one class compared to the other. As stated in my assumptions, we can say that the lab participants were paying attention, where as the online participants may or may not have been paying attention.

If the classes were balanced then simple approach may be to treat this problem as binary classification problem. Using something like a Support Vector Machine we could classify a given point as lab or online / paying attention or possibly paying attention based on their proximity to other data points. To do so we would need to have balanced classes otherwise the algorithm would have a high accuracy from just classifying everything as possibly paying attention as that is the most frequent class. There are two main methods of dealing with class imbalances, removing data samples and creating new data samples.

It was decided that creating new data samples would be best as there is not a whole lot of data to work with, so there would be a strong preference to keep the data we have. New points can be created by sampling from a distribution (reference) but here it was decided just to duplicate the samples as there was no discernible distribution of the data. Another method is to copy the points, altering them slightly, this was considered but not used in the end. Each lab study data sample was copied 40 times to even out the classes.

5.2 Features

This part of the pipeline refers to what features I am going to extract from the data. Features of data can be defined as 'attributes or interesting things from the data'. [reference] These will consist of both raw and created features, but what do I mean by this? Raw features will consist of the the number of mouse events recorded, while a created feature could be comparing the trace of users cursor data when using the program.

5.3 Machine Learning

Once we have insightful features from the data we can consider what machine learning algorithm would be most appropriate to use on the data. This will obviously be highly dependent on what form the final features are in. For example if the features are numerical values such as time taken to complete task and number of mouse events then an algorithm such as a Support Vector Machine would be a good choice. If the data is in the form of sequential data such as a list of all mouse events then something like a LSTM or RNN network would be best suited. If the features output was an image such as a trace of mouse position over time then a CNN could be a good choice as they're designed for image data. It is likely that text classification algorithms will be used when comparing the targets of mouse events. Comparing n-grams can be done with algorithms such as XYZ [reference]. Other text classification algorithms such as cosine similarity or sentiment analysis could also be used.

5.4 Labels

Lastly an important section of the pipeline, as it reflects the final outputs of the system. Labels will refer to which users are classified as paying attention and which users are not. A key aspect of this project will be semi-supervised learning. That is we have some data points we can confirm were paying attention, and others where they're level of attention was questionable. Once we have an algorithm that can classify some users as paying attention or not we can rerun the algorithm with these preliminary outputs as new training data. If this is done recursively then we can end up with a system that can split all data points into the 2 classes, perhaps with a degree of confidence given as a percentage.

6 Results

6.1 Table or graph of results

7 Conclusion

References

[1] Andrija Tomović, Predrag Janičić and Vlado Kešelj. "n-Gram-based classification and unsupervised hierarchical clustering of genome sequences". In: *Computer methods and programs in biomedicine* 81.2 (2006), pp. 137–153.

[2] Michael Collins. "Tagging with Hidden Markov Models". In: (2016).

[3] José Gordillo and Eduardo Conde. "An HMM for detecting spam mail". In: *Expert systems with applications* 33.3 (2007), pp. 667–682.

[4] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[5] Jaime Ferrando Huertas. *Generating synthetic data through Hidden Markov Models*. 2018.