Dissertation - Detecting User Engagement Using Mouse Tracking Data

David Saunders (910995)

September 2020

Abstract

This project explores the use of Hidden Markov Models to determine if users are paying attention during a crowdsourced study. The users are split into 2 distinct groups, online crowdsourced users, and lab study users. We propose that the lab study users were paying attention, and the crowdsourced users may or may not be paying attention. By using observed data recorded from the users mouse cursor we are able to model both groups interaction with the system with separate Hidden Markov Models. Some users interaction seem misplaced compared to their groups. These are reclassified with the aim of changing the groups of user from lab study or crowdsourced, to users paying attention and not paying attention. It was found that potentially 9.8% of turk users were paying attention, and 43% of lab study users may not have been paying attention.

Contents

1 Motivation

Crowd-sourcing marketplaces like Amazon's Mechanical Turk are popular services that provide a method for researchers to get participants to complete human intelligence tasks [1]. A common use for this technology is to label data for use in training for machine learning algorithms [2]. They can also provide a cheap, scalable method for scientists to gather responses in research. The level of user engagement, attention, and low quality responses can all be issues when gathering data from participants with distributed approaches [3]. The primary motivation of this project is to develop a system of potentially identifying if a crowdsourced user is paying attention during a task.

The use of gathering responses using Amazon's Mechanical Turk and other crowdsourcing alternatives are becoming prevalent across disciplines. It is commonly used in conducting clinical research [4] and it is estimated that almost half of all cognitive science research involves the use of crowdsourcing services to collect data samples [5]. The creation of an easy to implement method to measure user engagement would massively help researchers to increase reliability of their research.

Research has been conducted on how the accuracy and attention of crowdsourced tasks can be increased. Methods such as offering financial incentives [6] and engaging a users curiosity [7] have been found to motivate workers into performing better at crowdsourced tasks. Despite the research there is still debate as to which method is superior. If user engagement can be effectively identified then the best method of ensuring user engagement could be found using this method.

Measuring user engagement is well studied in the field of web analytics [8]. However the existing methods of reporting and analysing website data cannot be easily applied to crowdsourced tasks. Characteristics such as session duration and customer satisfaction are used as proxies for engagement. However a longer session doesn't necessarily mean a more engagement user and customer satisfaction is not applicable for crowdsourced tasks. Therefore existing solutions will not work and now methods must be evolved.

1.1 Background

The work in this project is build on an existing study; *"Risk fixers and sweet spotters: A study of the different approaches to using visual sensitivity analysis in an investment scenario"*[9].

The study was centred in the field of visualisation. The paper explored a new visualisation glyph that could be used help improve a users effectiveness at completing a task. The users task simulated an investment scenario, the aim was to maximise the return of a collection of stocks, while minimising the risk.

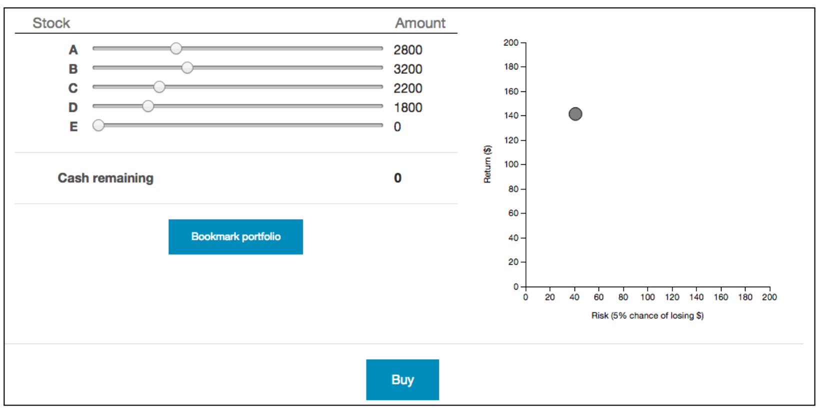


Figure 1: The user interface of the lab study, showing the risk and return of stocks to the left and allocation sliders to the right.

The experiment identified two users groups, risk fixers and sweet spotters, based on their response in a questionnaire. Figure 1 shows the interface of the lab study.

We divided the participants based on their response to a question about how they chose their final portfolio.

Table 1: Data collection methods used in the study [9].

Data collection method	Number of participants	Were participants paying attention?
Lab study	18	Yes
Crowdsourced task	370	Unknown

Table 1 shows the two different ways in which data was collected. Only 4.6% of the data was gathered in person, with the vast majority of responses being crowdsourced. The difference in number of participants shows one of the reasons crowdsourcing is popular. It is much easier to crowdsource responses than it is to organise an in person lab study.

1.2 Initial attempt

One naive assumption might be that a users attention level might be inferred from the time taken to complete the tasks. Exploration of the data showed that the division of users may not be so straightforward, shown in Figure 2.

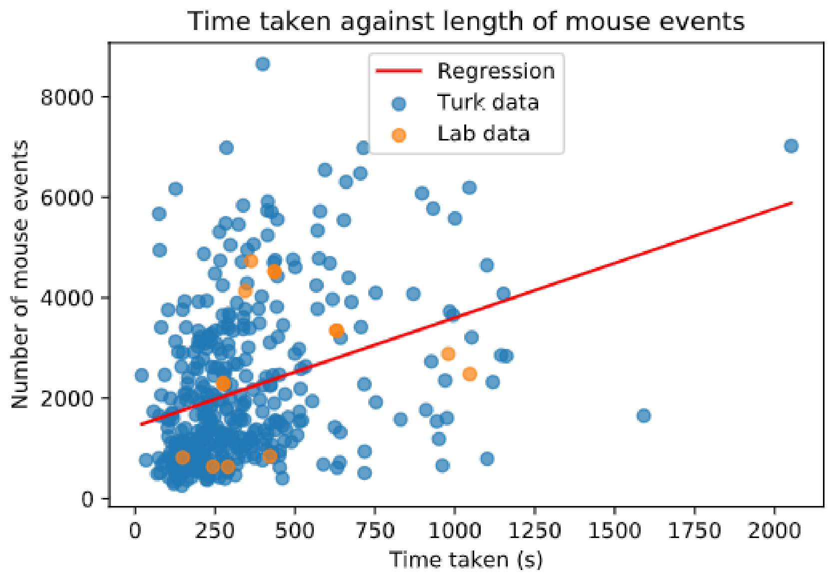


Figure 2: Scatterplot of users.

The lab and turk user classes cannot be separated by a simple metric of their data, such as length of events sequence and time to complete task. This rules out distance based machine learning approaches such as K Nearest Neighbours or a Support Vector Machine as there is no relationship there. It should be noted that length of events sequence and time are correlated with a pearson's correlation of 0.344.

As a rule of thumb we can say that correlations below 0.3 have no relationship, and values between 0.3 and 0.5 have a weak relationship [10]. This means the 2 features are weakly correlated, but correlated nonetheless. This means that one of these features may be a good substitute for the other and that having both may be redundant.

1.3 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 Eye tracking

Non-verbal information such as eye tracking may be used to detect user's level of engagement [11]. Vision is one of the most powerful human senses so it has the potential to give a good measure of user engagement. The methodology of eye tracking is that we move our eyes to focus on particular areas that we want to see in more detail, and divert our attention to that area [12]. Thus tracking a user's gaze can provide insight into which part of a system they're engaged with, and how much so.

Figure 3: Heatmap showing popular locations of users eyes on a webpage [13].

Eye tracking data can be used to show user interface elements that users focus their attention on as shown in Fig 3. From this researchers were able to predict the amount of attention elements of the page would receive. By observing what parts of an interface users are interacting with we can determine what a user is engaging with [13]. Eye-tracking has been used, and found success novel applications such as recording the engagement of users when playing a game. Tracking users eye movements helped game designers understand how users can recognise interactable game objects and could be used to investigate problematic game design issues [14].

Eivazi and Bednarik extracted features from a users eye tracking data to determine their cognitive state in a problem solving exercise [15]. Features such as "mean fixation duration" and "total path distances" were engineered, and users were split into classes based on their performance. Given a user feature set and their performance class it was able to classify their cognitive state during the task with a 87.5% accuracy with a support vector machine.

Szafir and Mutlu identified a plethora of verbal and non-verbal behavioural cues used by teachers in an educational setting, with gaze being identified as one of them [16]. The behavioural cues could not be recorded directly by a computer, instead EEG signals measured from a headset were used to measure engagement.

Eye tracking is not however a perfect solution and its limitations have been well documented. Track subjects eyes with a good degree of accuracy requires the use of expensive, intrusive equipment that frequently needed recalibrating [17].

2.2 Mouse cursor tracking

Research has also found that there is a correlation between a user's gaze and their cursor position. The position can be considered a "poor man's eye tracker" as it has been found that eye gaze match mouse position 69% of the time [18]. Mouse movement data can be collected without the drawbacks of eye tracking and with more automatic methods, meaning more data can be collected, and on a larger scale [19]. Therefore it can be said that mouse data can be used as a good alternative to eye tracking data.

By using mouse data it is possible to unobtrusively record a user's normal use of a web browser without disturbing their experience [20]. It has also been found that users tend to follow the text they are reading with the mouse cursor [21]. It can be determine what paragraph of a page was being read with an accuracy of 79% by using mouse cursor data [22].

Other methodologies explored ways of classifying user engagement from eye and cursor data, however it is also possible to predict users attention and user frustration in complex webpages [23]. Not all studies agree that mouse cursor is always a good approximation for eye data. Hauger et al found that distinct cursor behaviour exists depending on the task, and that the relationship between eye gaze and mouse position is more nuanced than measuring only mouse data [24].

2.3 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].

https://sci-hub.tw/10.2200/s00196ed1v01y200906aim006

SSL is the study of combining both labelled and unlabelled data to improve machine learning techniques.

Reference [25] as it looks to be massively influential with 4,000 google scholar citations. Mention semi-supervised SVMs here. And the abbreviation semi supervised learning (SSL)

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.)

Semi supervised learning has already been applied successfully to the field of user engagement. During a 2017 study video data of secondary school children was recorded during a set of tasks, with the aim of classifying pupils between the states of 'Engaged' and 'Disengaged' [26]. Rather than annotate all data by hand which would be very time intensive, 1000 frames of video were selected at random and hand labelled. This led to a split of data labelled 563 engaged and 437 disengaged. AU facial recognition features were extracted from the pupils faces in the video frames, and important features decided with a Principal Component Analysis method to keep features totalling 95% of the variance.

Their chosen method of semi supervised learning is a safe semi-supervised support vector machine. This variant of a semi supervised support vector machine finds a better separator when there is a small volume of labelled data, but a large volume of unlabelled data. The SSL model was trained with numbers of labelled data from 50 to 500, and and equal number of samples from both classes. The AUC metric ranged from 0.65 to 0.733 depending on the quantity of labelled dta used, outperforming a traditional SVM.

Unfortunately as shown in figure 2 the data here cannot be spatially separated and therefore any SVM would not be successful on the data. It is still useful however to see how semi-supervised learning has been applied to the field of detecting user engagement and how it has been successful.

2.4 N-Grams

This paper has a nice scientific explanation of n-grams [27]. Either cite this paper or more likely look at their reference for n-grams and cite that.

2.5 Hidden Markov Models

"Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process – call it X – with unobservable ("hidden") states. HMM assumes that there is another process Y whose behaviour "depends" on X. The goal is to learn about X by observing Y." Wikipedia

In the case of this project, the hidden states X would be the users thought processes including whether or not they're paying attention. The observations, Y, are the data that has been collected from the users such as mouse location over time and sequence of interface elements.

Paper Tom send me about HMM for text classification that I might be able to use [28].

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 [29]. 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 [30]. 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 [31].

Maybe this would belong under implementation?

Figure 4 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

To implement a HMM the python library HMMLearn was used [32]. The sequence data used to train the models was the order of mouse targets. The temporal aspect of the data was ignored, but earlier research showed that number of mouse events and time were heavily correlated. Therefore using just the order of mouse events should capture the temporal aspect of the data.

When implementing a Hidden Markov model on the actual data there are lots of parameters that must be considered. Given a sequence of input data the HMM model can calculate most of the parameters such as the transition matrix by using the Forwards-Backwards algorithm. The most prominent parameter that must be given is the number of hidden states to use.

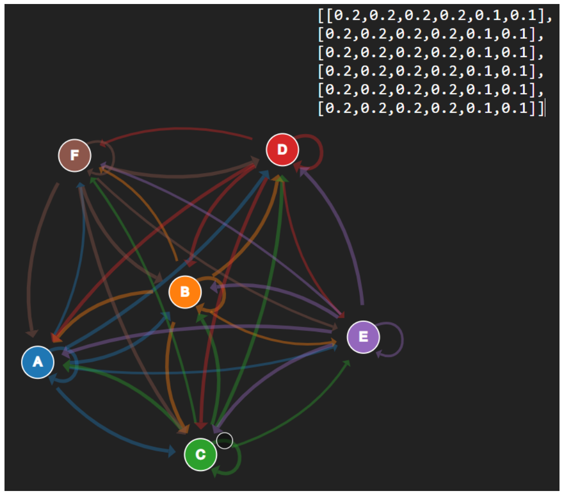


Figure 4: Diagram of a Markov Chain model representing a users mouse position

The paper [33] addresses this issue on similar data. The authors use a HMM to model animal movement behaviour, where they hope the hidden states would roughly relate to the behaviour of the animal. For example a 2 state HMM on this model may relate to ”foraging/resting” and “travelling”. Humans can be more complex than animals but we can think of potential states of being ”inquisitive” and ”bored” They trained models with 2, 3, 4, and 5 hidden states and compared the criterion’s of AIC, BIC, ICL. Similarity to evaluate Lab and Turk models models were trained with number of hidden states ranging from 1 to 11. Generally an increase in states should increase the effectiveness, however it will also take longer to train. All models used for this comparison was trained to 50 iterations for a fair comparison.

Figure 5 shows a different criterion of the total log likelihood of the training data of the model. This value is normalised by dividing by the number of data samples, to give a per-sample total likelihood. The evaluation of each HMM model does fluctuate based on the random state used, which is why there are drops in the evaluation criterion even when more hidden states are used. We can see that the criterion for both models stop improving after 9 states. Therefore a 10 state model was used.

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

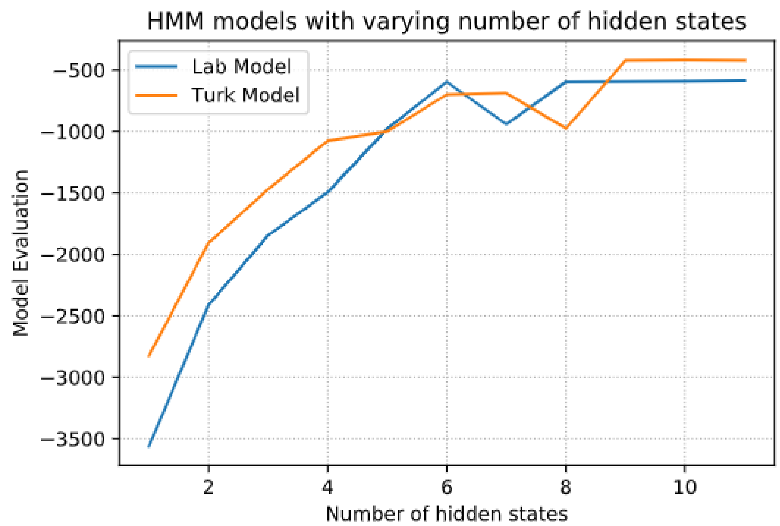


Figure 5: Line plot of evaluation of different HMM architectures.

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.

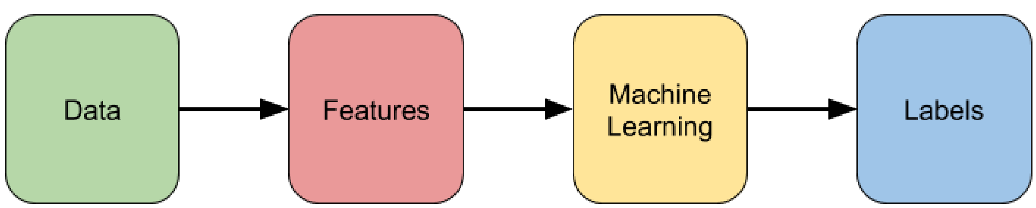


Figure 6: Diagram of the Data Pipeline of the project.

5.1 Data

The first component of the project has been completed, and data has been extracted form JSON format to a csv format.

Table 2: The first 5 records of results of the crowdsourced task.

event_type	target	time	x	y	step	turkId
mousedown	alloc-slider-1	0	477	405	1	A35YFAFWP33C70
mouseup	alloc-slider-1	0.111	478	405	1	A35YFAFWP33C70
click	alloc-slider-1	0.111	478	405	1	A35YFAFWP33C70
mousedown	alloc-slider-1	1.516	479	405	1	A35YFAFWP33C70
mousedirchange	alloc-slider-1	2.395	543	403	1	A35YFAFWP33C70
mousedirchange	alloc-slider-1	3.161	594	402	1	A35YFAFWP33C70
mouseup	alloc-slider-1	5.048	514	407	1	A35YFAFWP33C70
click	alloc-slider-1	5.048	514	407	1	A35YFAFWP33C70
mousedown	alloc-slider-2	5.461	494	441	1	A35YFAFWP33C70
mouseup	alloc-slider-2	5.513	494	441	1	A35YFAFWP33C70

Table 2 shows us the features of the data. The lab study data and the crowdsourced data have the same schema, except lab results have a different ID field.

The target field shows us which element in Fig 1 a participant is interacting with and event_type details the type of interaction. Time field shows the time taken in seconds since the first recorded mouse event. We can hypothesise that participants with a shorter time paid less attention than a participant who took much longer, thinking about their actions more. The x and y fields show the location of the mouse and step shows which stage of the task, from 1 to 5, a participant was in.

5.1.1 Data Manipulation

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 crowdsourced study. The purpose of the online study was to gather a larger amount of data that was possible to do so in person. Data was recorded in a JSON formate with lots of irrelevant and duplicate data relating to the users background and not their mouse movements. JSON is a form of unstructured data, meaning it is not fully structures but has some organisation to it [34]. To use the data in this project it first had to be processed into structured data. This was challenging and time consuming as Python's JSON library was unable to directly convert the data. This was because the mouse events were stored as a nested JSON dictionary and there were additionally errors with the way the data was logged causing it to be invalid JSON. After these challenges were addressed the data was converted and saved to a Pandas DataFrame, and then to a csv with over 100,000 lines. Errors were found in some of the data and so the final number of usable data is less than the figures shown in 2. Example errors include missing mouse events, or time to complete task being recorded incorrectly as 1.4×10^{12} seconds or $31,688$ years, obviously incorrect.

This leads to the problem of imbalanced data samples. After erroneous data was removed there were only 14 lab data samples and 461 online data samples. This means that there were over 30x 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. However due to the imbalances, alternative methods are required.

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

Now the real aim of the project can begin, attempting to classify users as paying attention or not. We will take the 2 HMMs and feed it in the existing mouse data. This will tell use the likelihood that that sequence of mouse events belongs to each class. If there are any points that appear to be outliers, then we can say that they were doing something differently to the majority of the turkers. If the same outlier also has a higher likelihood of belonging to the HMM trained on lab data then we can say that that outlier sample appears to belong to the wrong group. Or at least that outlier is doing something different to their peers, which may be a higher level of attention given to the task.

6.1 Table or graph of results

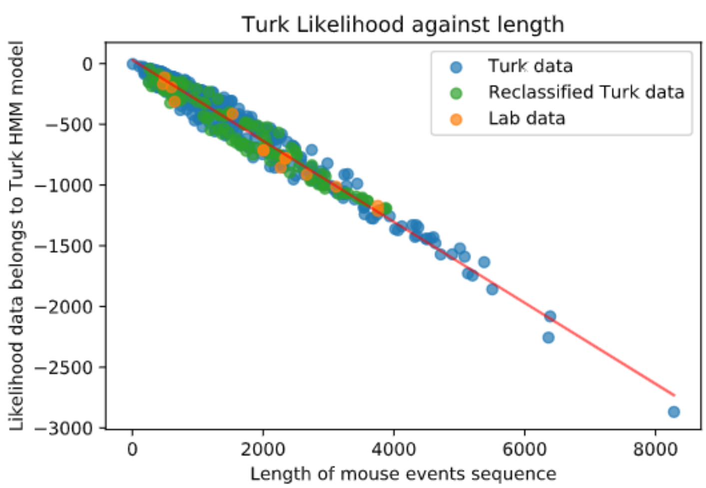


Figure 7: The comparative likelihoods of users classifiers with the Turk and Lab data.

Figure 7 shows how closely each users mouse sequences match the different HMM models.

The model likelihoods are the the log likelihoods that each user belongs to the lab or turk models. One way of attempting to classify users is to identify any users with a higher lab likelihood as being more similar to a lab user than a turk user , however all the points have a higher lab likelihood than turk likelihood shown by the black line y=x.

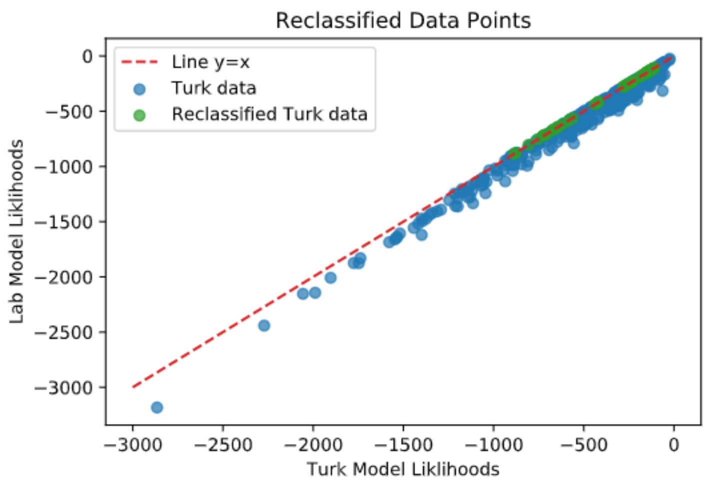


Figure 8: Plot showing the turk users that have been reclassified as behaving more like lab users.

Figure 8 shows which of the turk users are more similar to the lab users , than other turk users. This plot shows the same datapoints and axis , but the points above the line are recoloured yellow to show their difference from the actual lab data (green) and normal turk data (purple). 45 of the 461 (9.8%) turk users are above this threshold. Which seems like a reasonable percentage of the population to be paying attention. Additionally 6 out of 14 (43%)of the lab data samples has been reclassified by the algorithm as being more similar to the turk data.

The exact distribution of these points change based on the initial random state when training the model.

Figure 2 showed a naive simple attempt to separate the classes of turk users and lab users. It was concluded that a spacial based technique such as a SVM would be unsuitable as the data didn't seem to form any patterns or clusters. Figure 9 shows how the reclassified datapoints are not linearly separable. This proves that my initial hypothesis was correct and that data could not be reclassified purely by looking at simple feature of time taken and just number of mouse events.

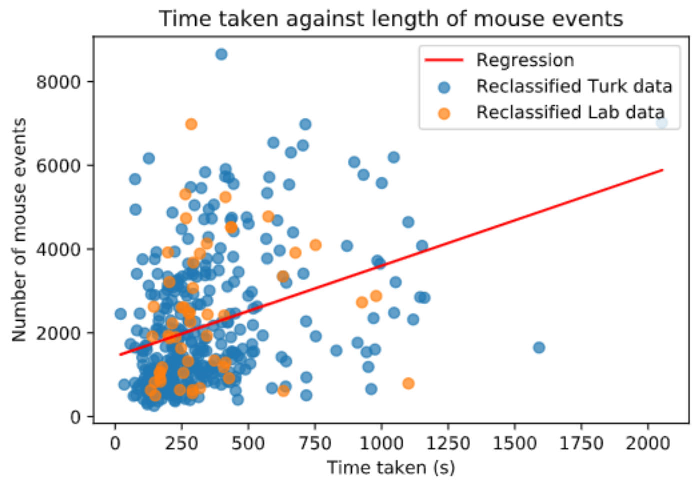
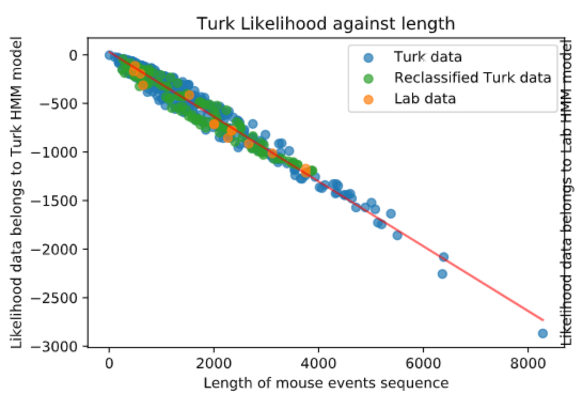


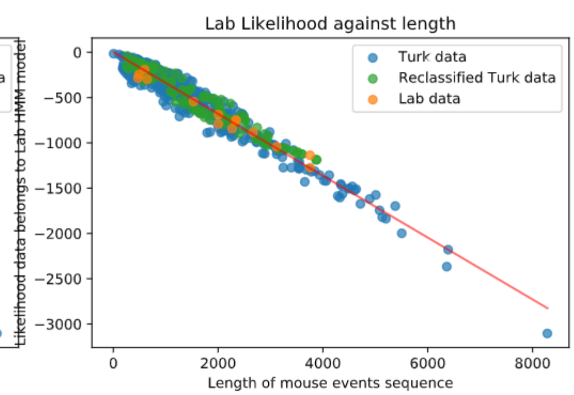
Figure 9: Plot showing features of the reclassified points.

6.2 Likelihood length correlation

Examination of the extreme points with the highest and lowest likelihoods revealed a potential correlation likelihoods and length of mouse events sequences. I decided to plot this data to understand if there was actually a link between these features.



(a) Turk HMM Model



(b) Lab HMM Model

Figure 10: Plots showing correlation between likelihood of different models and the length of mouse events.

Figure 10 show that the log likelihood from both models and length of a users mouse sequence are highly inversely correlated. The Lab model likelihood and length have a correlation of -0.982, and similarly the Turk model has a correlation of -0.976. A value less than -0.7 would indicate a strong negative correlation, therefore Figure 10 shows an extremely strong negative correlation [10]. Such a strong correlation could indicate the models are doing nothing, other that looking at the length of a sequence. However looking at the reclassified Turk data we can see that there doesn't appear to be any correlation between length, and whether the data has been reclassified. Therefore such a strong correlation between the length and likelihood is of no concern.

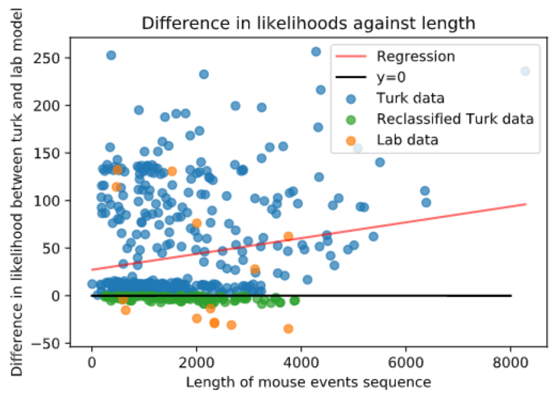


Figure 11: Plot showing difference in likelihood between the models against length of mouse events sequence.

Figure 11 removes the concern that length of mouse events sequence is the only feature being used in the models. The difference in likelihoods and length of mouse events sequence has a correlation of 0.176, meaning the relationship between the features is non-existant.

7 Conclusion

To evaluate the results I would not be too confident in my findings. The data was incorrectly labelled for the task I wanted to perform. Additionally the data was heavily imbalanced. All of the conclusions I have made from the lab data are relying on only a handful (13 maybe) of datapoints which is not statistically significant. Additionally there was not a lot of data from the other class of online data either. Typically data science and machine learning uses big data, where as this project used only 400ish records in total which came to only 500mb of data (MAYBE). Any bias in any of these original samples will be magnified as this was used to classify more points, which would again spread this bias. This would become a self fulfilling prophecy as more similar points would be labelled and spread the belief.

Any concern that the models have learned of a simple feature such as sequence event has been disproven , showing the reclassification algorithm has nonesuch simple relationship.

Biggest flaw of the project is the first assumption, that the lab people were all paying attention. While they were monitored and we can be sure they were not distracted by phones or televisions, many of them may have 'zoned out' and not have been given it their full effort and attention.

8 Future work

All the problems addressed in the conclusion have potential to be addressed in future work. As stated the main flaw of this project is that the label is not labelled for the task I am trying to perform. Therefore to fully evaluate any of these results would require further research with a properly labelled dataset.

Another area of future work would be to see if any of the methods developed here could be applied to other similar datasets. A kaggle crowdflower dataset seems like an idea candidate [35]. While it is still not labelled with attention and non-attention, it does contain mouse data of crowdsourced users. Additionally the dataset contains the users results from 3 different tasks, a users success at a task may prove to be a reasonable proxy for engagement.

References

[1] Gabriele Paolacci, Jesse Chandler and Panagiotis G Ipeirotis. "Running experiments on amazon mechanical turk". In: *Judgment and Decision making* 5.5 (2010), pp. 411–419.

[2] Joseph Chee Chang, Saleema Amershi and Ece Kamar. "Revolt: Collaborative crowdsourcing for labeling machine learning datasets". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 2334–2346.

[3] Panagiotis G Ipeirotis, Foster Provost and Jing Wang. "Quality management on amazon mechanical turk". In: *Proceedings of the ACM SIGKDD workshop on human computation*. 2010, pp. 64–67.

[4] Jesse Chandler and Danielle Shapiro. "Conducting clinical research using crowdsourced convenience samples". In: *Annual review of clinical psychology* 12 (2016).

[5] Neil Stewart, Jesse Chandler and Gabriele Paolacci. "Crowdsourcing samples in cognitive science". In: *Trends in cognitive sciences* 21.10 (2017), pp. 736–748.

[6] Chien-Ju Ho et al. "Incentivizing high quality crowdwork". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 419–429.

[7] Edith Law et al. "Curiosity killed the cat, but makes crowdwork better". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 4098–4110.

[8] Eric T Peterson and Joseph Carrabis. “Measuring the immeasurable: Visitor engagement”. In: *Web Analytics Demystified* 14 (2008), p. 16.

[9] Thomas Torsney-Weir et al. “Risk fixers and sweet spotters: A study of the different approaches to using visual sensitivity analysis in an investment scenario”. In: (2018).

[10] Diana Mindrila and Phoebe Balentyne. “Scatterplots and correlation”. In: *Retrieved from* (2017).

[11] Divesh Lala et al. “Detection of social signals for recognizing engagement in human-robot interaction”. In: *arXiv preprint arXiv:1709.10257* (2017).

[12] Andrew T Duchowski. “Eye tracking methodology”. In: *Theory and practice* 328.614 (2007), pp. 2–3.

[13] Georg Buscher, Edward Cutrell and Meredith Ringel Morris. “What do you see when you’re surfing? Using eye tracking to predict salient regions of web pages”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2009, pp. 21–30.

[14] Tony Renshaw, Richard Stevens and Paul D Denton. “Towards understanding engagement in games: an eye-tracking study”. In: *On the Horizon* (2009).

[15] Shahram Eivazi and Roman Bednarik. “Predicting problem-solving behavior and performance levels from visual attention data”. In: *Proc. Workshop on Eye Gaze in Intelligent Human Machine Interaction at IUI*. 2011, pp. 9–16.

[16] Daniel Szafir and Bilge Mutlu. “Pay attention! Designing adaptive agents that monitor and improve user engagement”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 11–20.

[17] Daniel C Richardson and Michael J Spivey. “Eye tracking: Characteristics and methods”. In: *Encyclopedia of biomaterials and biomedical engineering* 3 (2004), pp. 1028–1042.

[18] Lynne Cooke. “Is the Mouse a” Poor Man’s Eye Tracker”?” In: *Annual Conference-Society for Technical Communication*. Vol. 53. 2006, p. 252.

[19] Urška Demšar and Arzu Çöltekin. “Quantifying gaze and mouse interactions on spatial visual interfaces with a new movement analytics methodology”. In: *PloS one* 12.8 (2017).

[20] Jeremy Goecks and Jude Shavlik. “Learning users’ interests by unobtrusively observing their normal behavior”. In: *Proceedings of the 5th international conference on Intelligent user interfaces*. 2000, pp. 129–132.

[21] Chen-Chung Liu and Chen-Wei Chung. “Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages”. In: *IEICE transactions on information and systems* 90.10 (2007), pp. 1687–1696.

[22] David Hauger, Alexandros Paramythis and Stephan Weibelzahl. "Using browser interaction data to determine page reading behavior". In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer. 2011, pp. 147–158.

[23] Vidhya Navalpakkam and Elizabeth Churchill. "Mouse tracking: measuring and predicting users' experience of web-based content". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2963–2972.

[24] Jeff Huang, Ryen White and Georg Buscher. "User see, user point: gaze and cursor alignment in web search". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 1341–1350.

[25] Xiaojin Jerry Zhu. *Semi-supervised learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2005.

[26] Omid Mohamad Nezami, Debbie Richards and Len Hamey. "Semi-Supervised Detection of Student Engagement." In: *PACIS*. 2017, p. 157.

[27] 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.

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

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

[30] 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.

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

[32] Hmmlearn developers. *hmmlearn*. `https://github.com/hmmlearn/hmmlearn`. 2020.

[33] Jennifer Pohle et al. "Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement". In: *Journal of Agricultural, Biological and Environmental Statistics* 22.3 (2017), pp. 270–293.

[34] Jeremy Ronk. "Structured, semi structured and unstructured data". In: *Retrieved November* 21 (2014), p. 2015.

[35] Human Computation. *Workers Browser Activity in CrowdFlower Tasks*. Oct. 2017. URL: `https://www.kaggle.com/humancomp/worker-activity-crowdflower` (visited on 03/05/2020).