Basic experiment related to overconfidence detection method from PC interaction data

Picard Sébastien、　上村拓也、　鳥谷彰
富士通株式会社

1. Background

Overconfidence can lead to poor task performance or failure to achieve one's goals and can also prevent the implementation of optimal behavior change strategies. Overall, it can negatively impact a various range of industries from health promotion to fitness and exercise, to business and learning among others.

2. Problems with existing work

Typically, the way to approach the issue for the above-mentioned cases is as follows: analyze if the user has met their goal or not and compare that with a questionnaire about their expectation of success with regard to the goal [1], this allows to measure to what extent the user is overconfident. After that it is possible to take into account the overconfidence level for subsequent objectives and customize programs. However instead of waiting until the user has either achieved or failed at a first goal, it is desirable to detect overconfidence before adverse outcomes such as program abandonment or low achievement occur to be able to implement countermeasures.

3. Proposed method

We envisioned scenes where the user is interacting with a PC to carry out their work or to study. For such use cases, we used a definition of overconfidence similar to the over-estimation category from [2]. Specifically, we defined overconfidence as 1) the user will not be able to perform their task correctly or solve problems correctly 2) the user is confident of their future success and do not realize they will fail.

For this paper we focused on 1).　Our hypothesis is that we can infer whether the user will achieve their goal or not by detecting behavior that indicates the user is experiencing difficulty.

We focused on the aspect that when a user is operating a PC, if they hesitate because they are experiencing difficulty, the typing patterns will be slower than usual, also the time needed to make decisions will be longer (for example they will pause longer before clicking validation buttons) than usual. We conducted a principle experiment to verify the feasibility of our method. To reflect the above-mentioned behavioral hypothesis, we implemented a memory task and computed features such as follows: the duration of keystrokes, the duration between keystrokes, the duration hovering over a button before clicking it. We hypothesized that the more the user will experience difficulty recalling a combination from memory, the larger the feature values will be. Figure 1 illustrates this hypothesis.

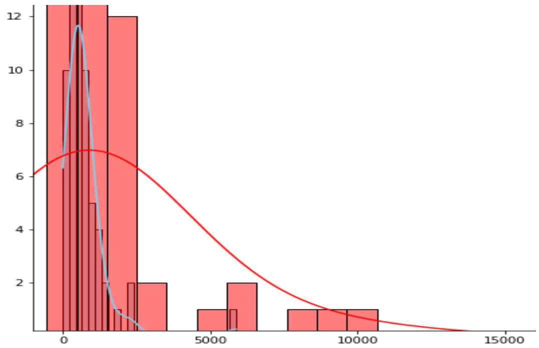

Figure 1): distribution of average duration in milliseconds between consecutive keystrokes. Blue line = 1st iteration (when the user is experiencing no difficulty), Red line = last iteration (when user cannot remember recall the combination)

In this experiment we started from simple combinations of digits, thus we could compute a baseline for each user (we assume the users experience no problem remembering the first easy combinations).

Based on the data of all the other participants in the experiment we built a model to assess whether the user will achieve their goal (until how many digits they can remember combinations) or not based on feature data relative to the baseline.

4. Experiment design

We devised a principle experiment to verify the feasibility of our method to detect whether a user operating a PC in an office setting with achieve their goal or not.

4.1 Task and experiment scenario:

- Display a random combination of digits for 5 seconds
- The user memorizes the combination and interacts with a custom UI requiring keyboard input and mouse movement before submitting their answer (this is called

an iteration for the rest of the paper)

- Starting from a combination with less than 5 digits, the combination grows in length with a 1-digit increment, either to the left or right to the previous combination
- This iteratively continues until the user either gives up or fails to remember up to 3 times (we call this series of iteration a sequence for the rest of the paper)
- Before the first combination is displayed the user is asked to input the expected length of the combination they can remember (We call this goal for the rest of the paper)
-

4.2 Example of features used as input for the model

Keyboard typing speed, average duration between keystrokes, time spent hovering on the submit button after typing the last digit.

5. Data collection

16 users responded to a pre-experiment survey and indicated their ascent to participate in the experiment. Explanations about how to install the program and run the user interface (UI) was provided to all participants. The task consisted of running the UI daily for 2 weeks (excluding weekends) and report their performance in a shared file so as to elicit overconfidence. 11 participants engaged with the program and we discarded the first sequence (used for practice). From the data obtained, we restricted the analysis to the iterations where the number of digits is lower that the goal, 66 sessions and 525 iterations.

6. Analysis

6.1 Features

We computed a battery of features meant to reflect our hypotheses stated in 3. Proposed method section. Those features included features computed from keyboard as well has mouse hovers over buttons to highlight the slowing of typing patterns expected to arise when hesitation due to inability to remember kicked in.

6.2 Normalization

Our hypothesis relies on comparing behavior with a normal level, we believe that changes in the above features are more meaningful than absolute values. To verify this, in the result section we compared performance for absolute values and data normalized as follows: a) the first 5 iterations of the second sequence (the first sequence being excluded from the data) is used to compute the minimum and maximum values independently for each participant. All features measuring durations (e.g. mean or standard deviation of the inter-keystroke duration) are normalized by subtracting the minimum and dividing by (maximum-minimum) independently for each user.

6.3 Model

To estimate whether a user is going to achieve or fail at their goal of remembering a certain number of digits, we used a statistical model which takes multiple features described in the previous section as input and outputs a binary label where 1 indicates the user will not meet their goal.

We used a random forest algorithm after early results (not reported in this paper) showing an improvement over other model tested (included KNN, logistic regression) and assumed that the in-built feature selection exhibited by random forest is responsible for the improvement of performance compared to other models. In this experiment we adopted a randomized search for the selection of hyper-parameters combined with a cross validation scheme. Further, to minimize the impact of overfitting due to personal differences we put emphasis in selecting folds such that data from participants cannot be both in the training and validation set. Additionally, we reported results where data from one user is exclusively used in the test set, this means that for user i in test, we trained the model based on the remaining 10 users, then we repeated for i = 1 to 11 and aggregated the results.

7. Results

From the computation of features related to the task where users remember a combination of growing length, we evaluated our method and reported results about the feasibility to predict whether the user will achieve their goal (meet their expectations) or not.

At first, we focused on each iteration separately. This task is rather ambitious but constitute the fundamental building block of our method.

We report the following results in figure 2): samples where the user does not meet their goal are labeled as 1 and samples where the user meets their goal are labeled as 0. The rows of the confusion matrix indicate the true label and the columns the predicted labels. We report the accuracy as well as balanced accuracy which is the average accuracy per class, we consider it is a better indicator in case of imbalanced data sets such as we have here. The models are also trained to maximize the balanced accuracy. For the rest of the paper, we report performance statistics rounded to the fist decimal. Figure 2a) shows the results for the basic model with no normalization. Figure 2b) show results with normalization where the user goal, iteration number of current number of digits are also added as features.

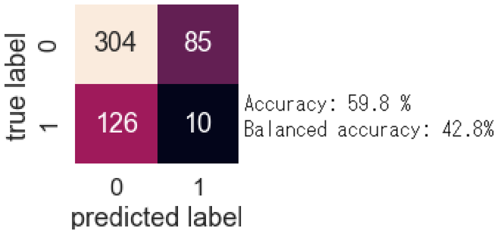

Figure 2a) confusion matrix without normalization

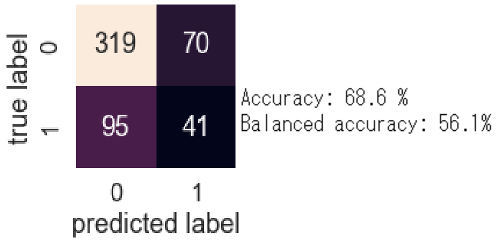

Figure 2b) confusion matrix with normalization

The results shown in figure 2a) confirm the feeling that the problem is extremely ambitious. Without normalization the balanced accuracy falls below 45% and reaches 56% with the normalization scheme. The improvement in balanced accuracy is substantial, especially in the latter case recall (41/136 = 30%) for class 1 is much higher compared with the former case (10/136 = 7.4%), however it is still much below 50%. For class 0, that is, when user will meet their goal, recall is 85 / 389 = 78.1% in the latter case and 319 / 389 = 82%. The results obtained here confirm the hypothesis that absolute feature values are less meaningful than relative values and we keep the normalization scheme for the rest of the analysis.

To verify our claims, we then evaluated the performance of our method to predict sequence labels, focusing on making predictions at an early stage (when the number of digits is lower than the goal). Due to the low number of sequences we decided to harness the iteration labels instead of computing features at sequence levels. However, with binary labels the information loss would be too significant, so a more fine-grained approach is preferable to predict sequence labels from iteration labels. Finally, for ease of comparison with the results described in figure 2, we output the sequence classification results in binary format.

The implementation steps are described here:

Step 1: modify the iteration model to output 4-class labels (L0: outperformed goal by more than 10 digits, L1: by 5 to 10 digits, L2: met goal or exceeded by less than 4 digits, L3:user didn’t meet their goal).

Step 2: obtain the 4-level labels for the first 5 iterations of each sequence and return the maximum value as intermediate result.

Step 3: return to a binary label by converting L0, L1, L2 to label 0, L3 to label 1.

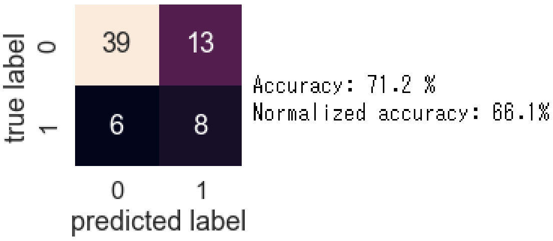

Figure 3): sequence level results

Figure 3) shows more promising results with a balanced accuracy reaching 66%, also recall for class 1 exceeds 50% with 8 / 14 = 57.1% but decreases for class 0 with 39 / 52 = 75.0%. The results obtained here show that it is possible to obtain better performance at sequence level by harnessing the results from the first few iterations.

We consider further tests with larger data sets are needed to confirm the reproducibility of the results and improvements in terms of prediction performance are required for various applications.

8. Conclusions and future work

Our experiment results indicate that it is possible to infer from PC interaction data the first part of our overconfidence definition, namely 1) the user will not be able to perform their task correctly or solve problems correctly.

We need to address the second part of our definition of overconfidence, namely 2) the user is confident of their future success and do not realize they will fail. We consider this may be done using a questionnaire, when the user is expected to be unable to reach their goal.

Overall, our vision is that overconfidence may be detected early enough to modify intervention strategies, benefit users and make significant impact for all sorts of applications. Future work includes various steps towards the realization of this vision, for example the implementation of the 2-step overconfidence detection in real-time and the evaluation of countermeasures for different applications.

9. References:

[1] Ifcher, John, and Homa Zarghamee. "Affect and Overconfidence: A Laboratory Investigation." Journal of Neuroscience, Psychology and Economics 7.3 (2014): 125-50. Print.

[2] Moore, Don A., and Paul J. Healy. "The trouble with overconfidence." *Psychological review* 115.2 (2008): 502