
EYE MOVEMENT CLASSIFICATION

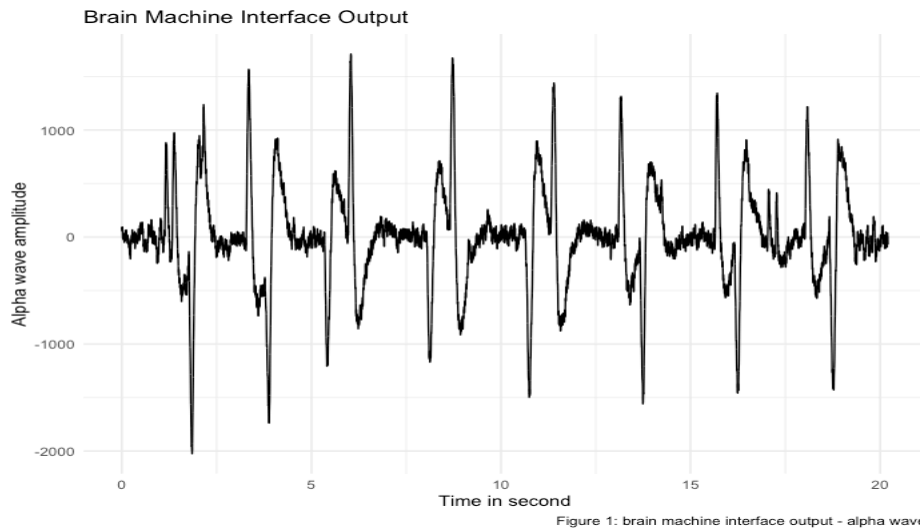
470539309

EXECUTIVE SUMMARY

This report uses the wave data generated by brain machine interface, aims to build a program which could detect eye movements and classify the left or right movement, with existing dataset and under the streaming condition. The standard deviation filter could detect eye movements and filter out non-movements. For existing dataset, the classifier random forest reaches an average accuracy of 0.985. Under the streaming condition, the classifier uses the sign of the signal with the largest absolute value. Among all the successful detections, the classifier's success rate is 96.58%, and it works better for shorter sequences.

BACKGROUND

Brain machine interface could acquire brain signals. It has electrodes attach to participants' head, which forms a closed circuit. When participants' eyes are close, brain machine interface could detect electroencephalogram, also known as alpha wave (Backyard Brains, 2014). When participants' eyes are open, eyeballs could be seen as dipoles. So that moving eyeballs left or right could change the direction of brain signals in the closed circuit. Brain machine interface could translate the signals. The output is simply the amplitude of alpha wave, with positive and negative values indicating the direction of the wave. Figure 1 gives an example of brain machine interface's output.



It is not difficult to distinguish the left eyeball movements, the right movements and no movement by observation. However, brain machine interface could only be useful when it could automatically detect the eye movement in a program. This report aims to build a program which could translate wave amplitude into eye movement. The program should also work in streaming conditions, which means, it should be able to detect eye movement simultaneously when signals come in.

CLASSIFICATION WITHOUT THE STREAMING CONDITION

METHOD

Before identifying the left or the right eye movement, intervals without any movements must be filtered out. By observing Figure 1, noticing that when there is no eye movement, the wave fluctuates around zero with low amplitude. In contrast, when there is a left or right eye movement, the wave deviates far from zero. Hence, standard deviation could be an appropriate filter to find the interval with eye movements. The threshold standard deviation is set at 650. The time interval to check is set at 1 second (10000 amplitude values within the interval), which is around the time needed for one eye movement. The interval is rolled by a step of 25 values each time (Appendix 1). For example, the first interval to check is from the first signal value to the 10000th value (0 second to 1 second), the second interval to check is from the 25th value to the 10025th value (0.00025 second to 1.00025 second). The standard deviation of each interval is recorded in a list (Appendix 2). Next, standard deviation for each interval is compared with the threshold 650. For intervals with higher standard deviation than the threshold, the median time is recorded in a list. As the rolling step is small, one movement is detected by several successive intervals. Sometimes, there are some small gaps between the intervals for one movement. To find different movements, firstly, all the gaps within the

list of median time are found, including small gaps within one movement and big gaps between different movements. Next, gaps bigger than average are recognised as gaps between different movements. Finally, intervals with no movements and intervals with eye movements are found (Appendix 3). The above steps are put into a function `identify_event_sd` for easier use.

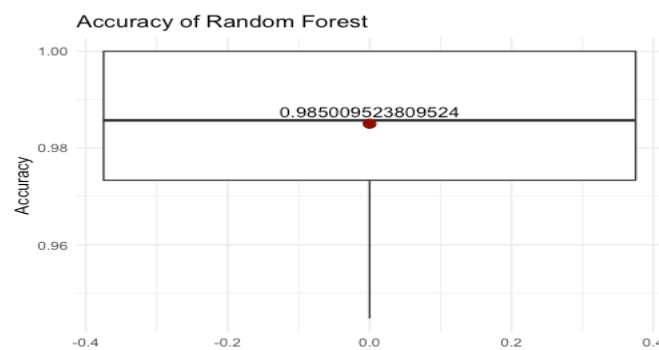
Function `extractSignal` could extract the signal values from intervals with eye movements (Appendix 4). Signals of all movements are extract from Zoe's short sequences are extracted and stored in `wave_seq_short`.

Function `features` could find features for each movement interval (Appendix 5). These features include mean, variance, number of zero-crossing points, lumpiness, etc.. Features of all movement intervals are extracted.

Random forest is chosen to be the classifier to detect the left or right eye movement. The features extracted from movement intervals are used as explanatory variables. The response variable should be either 'L' or 'R'. The known response variable is extract from Zoe's short sequences' file names. To estimate the accuracy of this classifier, a 5-fold cross validation is repeated for 50 times.

RESULT

Figure 2 shows the accuracy distribution of the 50 repeats. Random forest gives an average accuracy of 0.985. However, this classifier is only trained and tested using Zoe's short sequences. More details about this problem will be discussed in the DISCUSSION section.



CLASSIFICATION UNDER THE STREAMING CONDITION

METHOD

For the classification model without the streaming condition, as it is only trained by Zoe's short sequences, it is dangerous to use this classifier to detect others' brain wave, or longer wave sequences. Additionally, under the streaming condition, it is difficult to embed individual's differences into this classifier. Instead of using random forest, a more straightforward classifier is used to classify the left or eye movement. By observing the wave, the occurrence of eye movement could be recognised by large fluctuation. As an analogue of human's observation, the classifier recognises the occurrence by calculating the standard deviation. The type of movement could be recognised by observing the extreme point. The left eye movement is when the furthest extreme point from zero is negative, and the right eye movement is when it is positive. As an analogue, the classifier recognises the type of the movement by checking the sign of the value which has the largest absolute value.

Under streaming condition, signals come into the program successively. To save the computation time, a time interval of 1 second is checked every 0.1 second. For example, after the first 10000 signals come into the program, the first signal to the 10000th signal is checked, and then, the 1001th signal to the 11000th signal is checked. Additionally, a large step size of 1000 signals could effectively avoid the situation of one movement being recognised as several movements.

For each interval, the standard deviation is calculated. If the standard deviation is higher than a set threshold 500, the index (i^{th} number since the first signal coming in) of the first number in this interval is recorded into a list. If the standard deviation is lower than the threshold, the previous intervals with the index list which has more than one value are recognised as one movement. This ensures rare large fluctuation which is not an eye movement would not be recognised as a movement. Next, the classifier is applied to the movement to detect whether it is a left or right movement. Result of 'Left' or 'Right' is printed every time an eye movement is detected.

A while loop with the above steps and the classifier is used to simulate the streaming condition (Appendix 6).

RESULT

Zoe's wave file 'LLRRRLLL_z.wav' shows an example of the output. All the movements are successfully detected, and 7 out of 8 movements are correctly classified.

```
## [1] "0.9001 : 2.3001"
## [1] "Left"
## [1] "2.8001 : 4.3001"
## [1] "Left"
## [1] "5.1001 : 6.5001"
## [1] "Right"
## [1] "7.8001 : 9.2001"
## [1] "Right"
## [1] "10.1001 : 11.8001"
## [1] "Left"
## [1] "12.8001 : 14.2001"
## [1] "Left"
## [1] "15.3001 : 16.7001"
## [1] "Left"
## [1] "17.8001 : 19.2001"
## [1] "Left"
```

Figure 3 shows the testing wave and detection and classification result. The 5th movement should be a right movement instead of left.

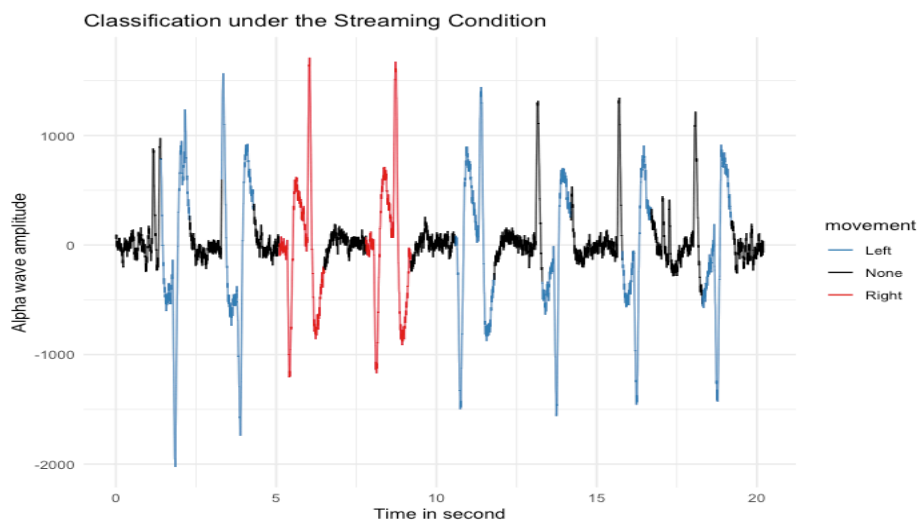


Table 1 shows the result under the streaming condition after testing both Zoe's and Louis' short, medium, long sequences. It has 87% movements detected correctly. Among the correctly detected movements, 97% are classified correct.

Result of Detection and Classification under the Streaming Condition		
All Movements	Success Detection	Success Classification
304	263	254

Table 1: Result summary of detection and classification under the Streaming Condition

Figure 4 shows an unsuccessful detection. The first and the last movement are not detected, and the detected movement's interval does not include the entire movement. The set threshold of standard deviation, the interval length and step size affect the detection of movement.

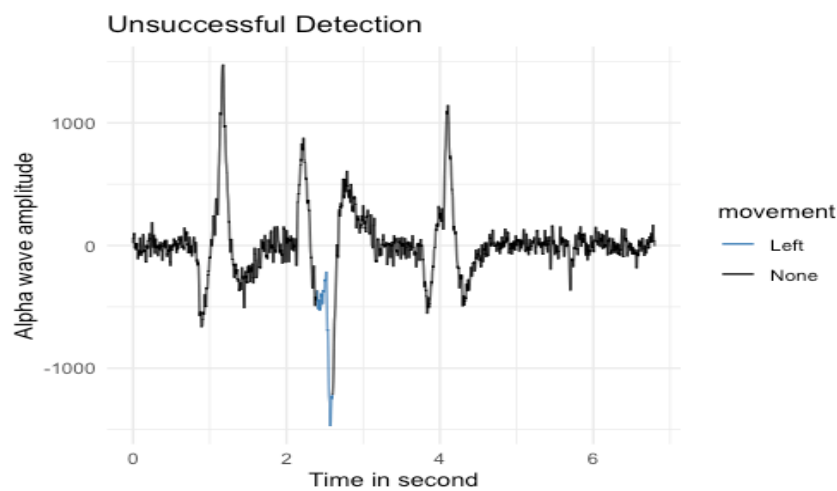


Figure 4: an example of unsuccessful detection

The classifier classifies the left or right detection by finding the largest absolute value. However, sometimes a left movement could have an absolute value of its smallest value smaller than its largest value, vice versa. Figure 3 gives an example of such issue, when the 5th movement should be right instead of left. To improve the classification performance, a classifier which compares the order of its largest value and smallest value could be more accurate. However, such a classifier will require a better detection method. As our current classifier reaches 96.58% success rate, another classifier will not be considered in this case.

DISCUSSION

RANDOM FOREST CLASSIFIER

The random forest classifier is also tested in the while loop under the streaming condition. It gives good results for Zoe's short sequences. However, when the test wave is Zoe's medium sequence or long sequence, it classified many right eye movements as left eye movements. The random forest classifier works worse with longer sequences. Additionally, the random forest classifier works worse on all of Louis' short, medium and long sequences. This is because the trained dataset is only from Zoe's short sequences. The eye movement pattern is different from person to person. And the longer the sequences, the worse this classifier performs. As a result, a random forest is not a good classifier for the streaming condition. The sign classifier is more general and suits better for the streaming condition.

IMPACT OF SEQUENCE LENGTH ON CLASSIFICATION

Table 2 shows result of different sequence lengths under the streaming condition. When wave sequences get longer, the classifier performs worse.

Result of Detection and Classification by Sequence Length				
	All Movements	Success Detection	Success Classification	Classification/Detection
Short	144	109	107	98%
Medium	80	75	74	99%
Long	80	79	73	92%

Table 2: Result of detection and classification by length under the Streaming Condition

PERSON TO PERSON VARIATION

Table 3 shows result from different person. Louise has a smaller detection rate which could because of the smaller standard deviation of his eye movements. This could also be observed from Figure 4. However, among all the detected movements, Zoe has a smaller successful classification rate. This is because, for example, according to Figure 3, the positive largest value from Zoe's right movements sometimes is larger than the absolute value of the negative smallest value.

Result of Detection and Classification under the Streaming Condition						
	Zoe			Louis		
	All Movements	Success Detection	Success Classification	All Movements	Success Detection	Success Classification
Short	72	65	63	72	44	44
Medium	40	35	34	40	40	40
Long	40	39	33	40	40	40
Total	152	139	130	152	124	124

Table 2: Result of detection and classification by length under the Streaming Condition

CONCLUSION

This report builds a program to detect eye movement and classify the left or right movement. Standard deviation is used as the detection filter. Random forest is the classifier for the existing dataset, which has an average accuracy of 0.985. The sign of the signal with largest absolute value is the classifier under the streaming condition. The detection rate is 87%. Successful classification rate among detected movements is 97%. The classifier works better with shorter sequence. The wave signal is different from person to person, in terms of movement amplitude, standard deviation and movement pattern.

REFERENCE

Backyard Brains.(2014). *Alpha Waves of the EEG*. Retrieved from <https://www.youtube.com/watch?v=Y1tCV9Qopv4>

Jean Yang. (2020). *DATA3888 Data Science Capstone. Lecture 1*. The University of Sydney, Sydney.

Shih, J. J., Krusienski, D.J. & Wolpaw, J.R.. (2012). *Brain-Computer Interfaces in Medicine*. Mayo Clin. Poc.

APPENDIX

Appendix 1. Rolling intervals

```
x = max(xtime) - windowSize
indexLastWindow = max(which(xtime <= x)) + 1
ind = seq(1, indexLastWindow, by = downSampleRate)

timeMiddle = xtime[ind] + windowSize/2
testStat = rep(NA, length(ind))

for (i in 1:length(ind)) {
  Y_subset = Y[xtime >= xtime[ind[i]] & xtime < xtime[ind[i]] + windowSize]
  testStat[i] = sd(Y_subset)}
```

Appendix 2. Finding standard deviations

```
predictedEvent = which(testStat > thresholdEvents)
eventTimes = timeMiddle[predictedEvent]
```

Appendix 3. Grouping movements

```
gaps = which(diff(eventTimes) > mean(diff(eventTimes)))
noiseInterval = rbind(
  c(range(xtime)[1], min(eventTimes)),
  cbind(eventTimes[gaps], eventTimes[gaps+1]),
  c(max(eventTimes), range(xtime)[2]))

eventsInterval = cbind(noiseInterval[-nrow(noiseInterval),2],
  noiseInterval[-1,1])
```

Appendix 4. Extracting signal function

```
extractSignal = function(limits, seq, xtime) {
  index = (xtime > limits[1]) & (xtime < limits[2])
  return(seq[index])}
```

Appendix 5. Extracting features function

```
features = function(alist) {
  cbind(tsfeatures(alist, c("acf_features", "lumpiness", "flat_spots", "crossing_points")),
    tsfeatures(alist, "max_kl_shift", width=48),
```

```
tsfeatures(alist, c("mean", "var"), scale=FALSE, na.rm=TRUE),
tsfeatures(alist, c("max_level_shift", "max_var_shift"), trim=TRUE))}
```

Appendix 6. Streaming

```
window_size = 10000
step_size = 1000
last = length(testWave)-window_size+1
i = 1
stamp = NA
happen = FALSE
m=1
movement = rep('None',length(testWave))

while (i < last) {
  window = testWave[i:(i+window_size-1)]
  if (sd(window) > 500) {
    stamp[m] = i
    m = m+1
    happen = TRUE}

  else if (happen == TRUE) {
    if (min(stamp) != max(stamp)) {
      event = testWave[min(stamp):(max(stamp)+window_size/2)]
      LR = sign(unique(event[abs(event) == max(abs(event))]))
      print(paste(time[min(stamp)], ': ', time[max(stamp)+window_size/2]))

      if (LR == 1) {
        movement[min(stamp):(max(stamp)+window_size/2)]='Right'
        print('Right')}

      else {
        movement[min(stamp):(max(stamp)+window_size/2)]='Left'
        print('Left')}

      m = 1
      stamp = NA
      happen = FALSE}}
  i = i+step_size}
```

Appendix 7. Result of streaming

Result of Detection and Classification under the Streaming Condition									
	Zoe			Louis			Total		
	All Movements	Success Detection	Success Classification	All Movements	Success Detection	Success Classification	All Movements	Success Detection	Success Classification
Short	72	65	63	72	44	44	144	109	107
Medium	40	35	34	40	40	40	80	75	74
Long	40	39	33	40	40	40	80	79	73
Total	152	139	130	152	124	124	304	263	254