Improving Movement Onset Detection in EEG-based BCIs with 'Weakly' Supervised Learning

Bachelor Thesis in Artificial Intelligence

Submitted to the Faculty of Social Science of the Radboud University, Nijmegen

Guus van der Ham

Supervisor:

Ceci Verbaarschot

Nijmegen, The Netherlands

July 9th, 2019

Abstract

As EEG based BCI systems that are expected to function in everyday situations are developed, the challenge to deal with noise due to the environment or the subject itself becomes more important. In this project it was explored whether the 'weakly' supervised learning techniques for movement onset detection with an EEG-based BCI used by Awwad Shiekh Hasan et al. might improve the classification accuracy in the experiment conducted by Verbaarschot et al. This experiment, in which participants played a self-paced BCI-game, was carried out during the InScience festival; a noise-rich ecologically valid environment. It was found that methods proposed by Awwad Shiekh Hasan et al. performed better than the linear classifier Verbaarschot et al. used however both were not very accurate in detecting movement onset.

Introduction

A Brain-Computer Interface (BCI) is a way for a human to communicate with a device using nothing but brainwaves. BCIs generally can be divided into two classes based on how they work: synchronous and asynchronous. In synchronous BCIs it is known beforehand when the onset of some mental activity is going to take place and so the system can analyse the data in predetermined time intervals. An asynchronous BCI, often also called a 'self-paced' BCI, allows the user to take action when desired(Mohammadi, Mahloojifar, Chen, & Coyle, 2012). In this case the system needs to detect when the user is in the idle state and when he or she switches to taking action.

When a person plans to move it is possible to detect certain brainwaves indicating this intention about 1.5 seconds before this person reports being actually aware of having made the decision to move(Libet, Gleason, Wright, & Pearl, 1983). This suggests the brain has a certain preparatory phase before the onset of a movement. The brainwaves connected to this phenomenon are the readiness potential (RP)(Lew, Chavarriaga, Silvoni, & Millán, 2012) and the event-related-desynchronization (ERD) that are visible at 8-30Hz across the motor cortex(Pfurtscheller & Lopes da Silva, 1999). The goal of movement onset detection is to identify this preparatory phase in order to predict the presence of an intention to move.

One of the challenges with EEG-based BCIs is that they are vulnerable to all kinds of electromagnetic noise caused internally or externally. Sources include devices we frequently use in our daily lives like smartphones and computer screens(Repovš, 2010) but also the blink of an eye or a distracted participant can disturb the EEG signal making it harder to detect a mental state. This is especially the case when this mental state is not time-locked. However, if we want to be able to use EEG-based BCIs in everyday life we need to build systems that are robust and can function under circumstances where noise from all kinds of sources is present.

Applications of movement onset detection include hands-free input devices and BCI-games like Mattel's Mindflex[1]. Furthermore, it may be used to improve the accuracy of neuroprosthetics(Müller-Putz, Scherer, Pfurtscheller, & Rupp, 2005) and the recovery for stroke patients(Ang & Guan, 2013). One thing these applications all have in common is that they have to function in everyday situations and have to deal with the noise that comes with it. For this reason it is important to research BCI techniques in ecologically valid experiments.

One of such experiments was conducted by Verbaarschot et al. during the InScience Festival[2]. Using an EEG-based BCI Verbaarschot et al. were able to predict movement onset in a game called 'Flip-that-Bucket'(Verbaarschot, Gerrits, Haselager, & Farquhar, 2019). In this game the goal of the participants is to beat the robot in a slime-bucket challenge. During the game slime is collected in a single bucket. Both the participant and the robot can push on a button that flips the bucket of slime over the opponent's head. The goal is to spill as much slime over the other as possible. However, making use of the EEG data the robot is able to detect the player's intention to push the button and it will try to press it earlier.

To perform such predictions in self-paced BCI's the system needs to know the transition onset from idle/baseline state to movement. This may be hard to detect online as the transition happens gradually(Awwad Shiekh Hasan & Gan, 2010). According to Awwad Shiekh Hasan et al. unsupervised learning may provide a possible answer to handle this lowly separable data. They proposed an "unsupervised" method that was able to improve the movement onset detection accuracy in an EEG-based BCI experiment. However, this "unsupervised" method turned out to be 'weakly' supervised in certain steps. This will be examined in more detail later. They compared this method

[1] https://store.neurosky.com/products/mindflex
[2] https://www.insciencefestival.nl/en/

with a fully supervised variant in an experiment where 5 participants performed a hand movement on their own pace 40 times with at least 4 seconds between every movement. In this way a 'clean' dataset can be built with a clear distinction between the baseline and movement class. However, it is not so much an ecologically valid situation.

In this research the 'weakly' supervised method proposed by Awwad Shiekh Hasan et al. was applied on the dataset collected by Verbaarschot et al. to find out whether the proposed method could improve movement onset detection in the (ecologically valid) BCI-game 'Flip-That-Bucket'.

Methods

Participants

Verbaarschot et al. tested 41 subjects at the InScience festival in Nijmegen, the Netherlands of which 9 were later excluded as they did not follow instructions correctly. In the final analysis an additional subject was excluded by Verbaarschot et al. This subject was therefore also not included in this research as it would not have any comparative value. The final number of subjects is 31.

Apparatus

The data was collected using the TMSi Porti system[3] with water based electrodes. The sampling rate was 512Hz and the electrodes where placed at Fp1, Fp2, F3, Fz, F4, C3, Cz, C4, P3, Pz, P4, POz, TP9 and TP10 according to the International 10/20 system[4]. Furthermore, muscle activity was recorded using two EEG electrodes in a bipolar pair on the wrist and centre of the right forearm (flexor pollicis longus).

Task

The subjects played several rounds of Flip-that-Bucket in 4 blocks: a practice block of 3 trials, a training block of 60 trials, a hidden validation block of 15 trials and a test block of 60 trials.

Dataset

The trials were labelled with 3 types of events: 'player act' where the subject acted before the robot did, 'robot act with intent' where the robot acted and the subject had an intention to move and 'robot act without intent' where the robot acted without the subject having an intention to move. Not all trials were used. Verbaarschot et al. report a big response of the subjects when the robot acted first. According to them this might be related to an error potential or it might reflect the subject's surprise or frustration after losing. Due to this big response the movement onset might have been interrupted or disturbed in the trials where the robot acted first. For this reason those trials have been excluded leaving only trials where the player acted first.

Data Analysis

The methods of Awwad Shiekh Hasan et al. can be divided into six general steps: pre-processing, feature extraction, feature selection, model-building, classification and onset detection. Following is a detailed explanation of each of these steps.

1.Pre-processing

First the continuous EEG- and EMG-data was sliced into segments of 6 seconds around a 'player act' event, specifically 4 seconds before and 2 seconds after such event. Then the data was pre-processed using the Common Average Reference(CAR) method. Furthermore, the data was demeaned, detrended and the bad channels and bad trials where the power differed more than 2 times the standard deviation

[3] http://www.tmsi.com/products/porti/
[4] http://www.mariusthart.net/downloads/eeg_electrodes_10-20.svg

were removed. After these steps for each of the participants on average 67 good trials were left with min. 25, max 107 and std. 18. No channels had to be removed.

2.Feature Extraction

For every trial the EEG-data was sliced into windows of 1 second with each window shifting ⅛ of a second. With a sampling rate of 512Hz this makes for a window of 512 samples shifting 64 samples a time and a total of 40 windows per trial. For all 12 channels the Power Spectral Density(PSD) of each window was computed using the Thomson Multitaper Method(Thomson, 1982). Only a selection of 10 frequencies in the range of 8-26Hz sampled at 2Hz was used. To reduce dimensionality at an early stage the decision to deviate from Awwad Shiekh Hasan et al.'s method was made. Awwad Shiekh Hasan et al. also include frequencies in the range of 28-45Hz, sampling at 3Hz. However, as expected, almost no activity was seen in this range as the expected Event-Related Desynchronization takes place in the alpha and beta band (8-30Hz)(Bai et al., 2011). This phenomenon can be seen as a drop in power in said range around the button press (t=0). This was seen especially well when the power was averaged over all trials of subject 1 in channel C3.

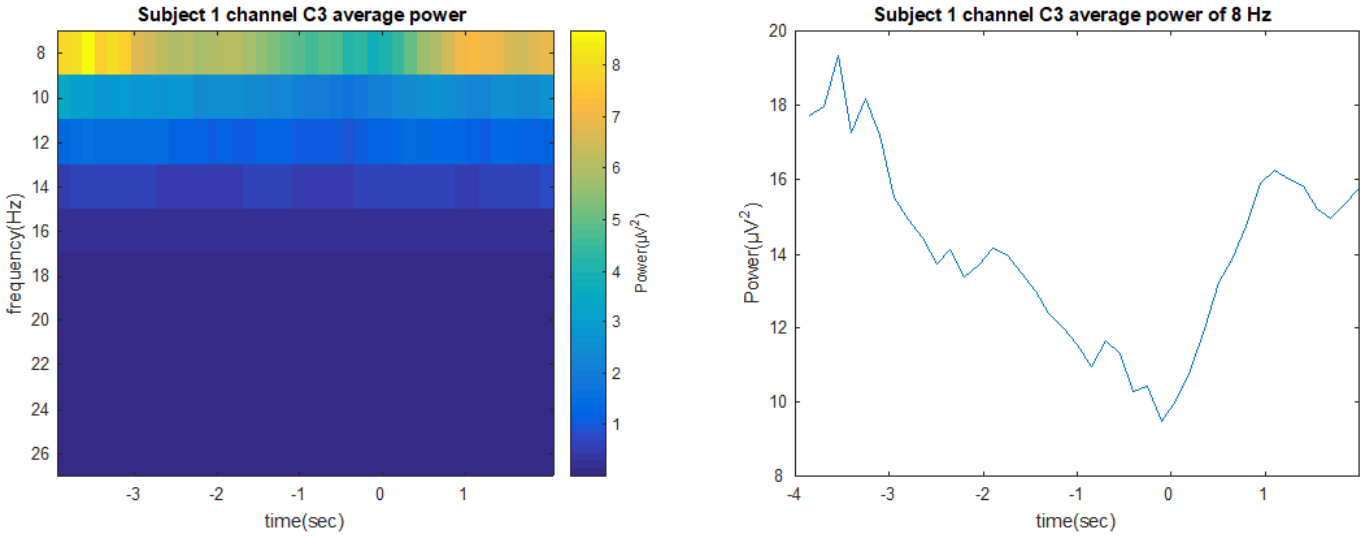

Figure 1: Observed power drop in alpha and beta band and at 8Hz

3.Feature Selection

To perform feature selection the data was labelled into a 'movement' and 'baseline' class based on EMG activity. The movement class was further subdivided into a 'preparation-', 'execution-' and 'after-execution' class. First the EMG onset and offset of each trial was computed using a script by Coghlan (Coghlan, 2006) based on methods by Hodges and Bui(Hodges & Bui, 1996). For this method first the mean and standard deviation of a resting phase is determined. The threshold for EMG-activity is set to the mean + 1*standard deviation of the resting phase. The EMG-onset is determined by the first time the average activity in a window of 50ms exceeds the threshold and the EMG-offset by the first time the activity dives below this threshold.

When the result was undetermined or not around the time the button was pressed the average onset and offset of the trials was taken. Using the EMG onset and offset times the data was labelled 'preparation' (p) from 1,5 seconds prior to onset until onset, 'execution' (e) from onset until offset, 'after-execution' (a) from offset until 1,5 seconds after onset and the remaining samples were labelled as 'baseline' (b). In figure 2 an example of such labelling is shown on the average EMG-signal over all subjects, with the mean onset time (3,78sec) and mean offset time(4.65sec) over all subjects. Finally, each window was labelled with the label of the most common class in that window. As these labels are used in selection of the best features this method as proposed by Awwad Shiekh Hasan et al. can not be referred to as unsupervised learning. 'Weakly' supervised learning was found to be more accurate in this situation.

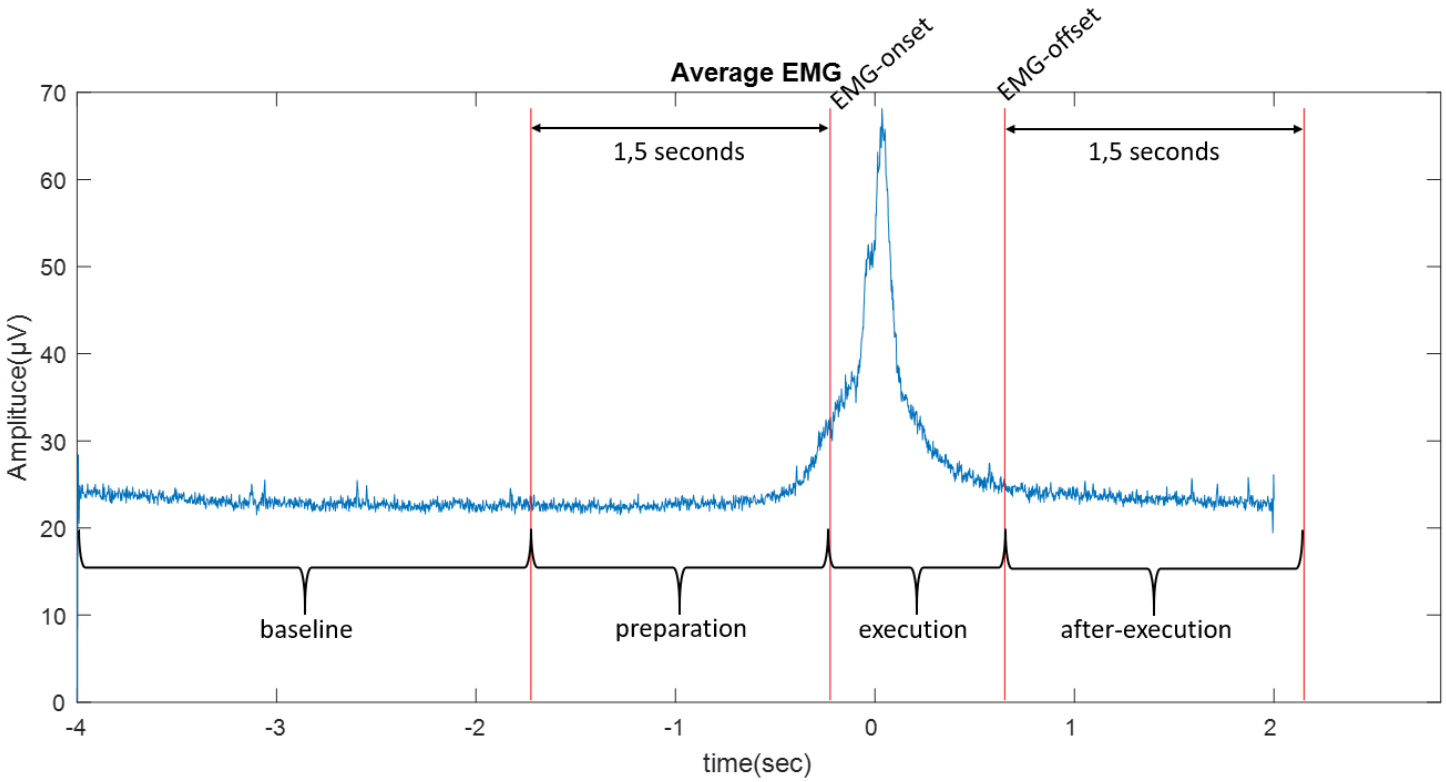

Figure 2: Average EMG and datalabelling

Resulting from the feature extraction there are 12 channels x 10 frequencies making a total of 120 features. As Gaussian Mixture Models(GMM) do not perform well on high dimensional data(Zhao, Shrivastava, & Tsui, 2018) feature selection was done to select a number of features that separate data in the preparation and execution class best from the other classes. To do so Awwad Shiekh Hasan et al. compute the Davies Bouldin Index(DBI) for every feature. The DBI is an index that evaluates how well the data is clustered into the given classes. This is done in the following steps:

First A_i is computed which is the centroid of class i. This boils down to the mean value of that feature, where, to clarify, a feature is the PSD value of a specific frequency at a specific channel.

Then S_i is computed for every class i, which is in essence the within class scatter:

$$S_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^p \right)^{\frac{1}{p}}$$

Where T_i is the number of datapoints in class i, X_i is a datapoint assigned to class i and A_i the centroid of class i. In this case we take $p = 2$ making this essentially a function of the Euclidean distance between datapoint $X_i$and cluster centroid A_i.

Then M_{ij} is computed for every combination of class i and j where $i \neq j$. This is in essence the between class scatter:

$$M_{ij} = \left(\sum_{k=1}^{n} |A_{k,i} - A_{k,j}|^p \right)^{\frac{1}{p}}$$

Where again $p = 2$ and $A_{k,i}$ is the k[th] dimension of the centroid of cluster i. As datapoints in this case are one dimensional values ($n = 1$), the computation of M_{ij} boils down to the absolute difference between the two cluster centroids.

Then R_{ij}, which combines the within- and between class scatter, is computed as follows:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$

Finally the DBI for a certain feature is defined as follows:

$$DBI = \frac{1}{C} \sum_{i=1}^{C} D_i$$

where

$$D_i = max_{j \neq i} R_{ij}$$

and C is the number of classes.

However, Awwad Shiekh Hasan et al. select N features based on the DBIs *"that maximise the separability of ''preparation'' against other subclasses"* and N features *"that maximise the validity of ''execution'' against other classes"*. This was interpreted as them not using the final DBI for a feature but the D_i value for preparation and execution. The DBI of a feature does not evaluate the separability from a specific set of classes but evaluates its separability from all other classes. The D_i value however does say something about the separability of one specific class against the others, the mean of all of those D_i values makes the DBI score for a feature. In this case $D_{preparation}$ and $D_{execution}$ were used to select the best N features to build the model where N is always a multiple of 2.

4.Model Building

The type of model Awwad Shiekh Hasan et al. built is a Gaussian Mixture Model(GMM). In this model it is assumed that the data is generated by K normal distributions where K is finite. These normal distribution are referred to as components. Naturally these components have a mean μ_k and a variance Σ_k but also a mixing coefficient π_k. One could say this mixing coefficient determines how 'dominant' a certain component is in generating the data relative to the other components. Classification using a GMM happens by computing the probability that a datapoint is generated by a certain component which in turn has a probability that is belongs to a certain class.

For building and evaluating the model and eventually testing the methods as a whole, 10-fold cross validation is used twice. Once for building and selecting the best model, optimum number of components K and optimum number of features N. And once for evaluating the whole system of onset detection. First the data is divided, trial by trial, into 9 parts train data and 1 part test data. To avoid confusion this test data is from now on referred to as 'evaluation' data. The evaluation data is used to

evaluate the method as a whole in the final step. The train data again is divided, trial by trial, into 9 parts train data which is used to build the model and 1 part test data to test the classification accuracy of the model.

Awwad Shiekh Hasan et al. experiment with the number of components but limits K in the range of 4-20 and the number of features N in the range of 2-100 as the performance could be severely affected by the high dimensionality of the data. According to them using more features could cause overfitting as it becomes harder to build accurate probabilistic models with such high dimensional data. However, when reproducing this it was found that the GMM often would not settle and produced errors related to the high dimensionality of the data already when N was larger than 10. Furthermore there were warnings for a so called variance floor when K was larger than 8. When the covariances of a GMM are too low it is a sign that overfitting could be taking place. For this reason K was taken in the range of 2-8 and N was taken in the range of 2-10. This does not pose a problem for the comparability to Awwad Shiekh Hasan et al.'s methods as they also report optimum parameters within or close to this range. For comparison these are also reported in the discussion&conclusion section.

To find the optimum parameters for the model the train data was taken and for every possible combination of K and N a second round of 10-fold cross validation was used to find the best performing model. First a model was fitted to the train data. This model was trained using the Expectation-Maximization method. Then the model was used to classify the test data. Using the class labels as defined in step 3 the classification accuracy was computed. Finally the best performing model was further used for classification.

5.Classification

As explained the data is classified using probability distributions. To start with, it is assumed that the data is modelled by the following probability density function:

$$P(x) = \sum_{k=1}^{K} \pi_k \, N(x|\mu_k, \Sigma_k)$$

Where K is the number of Gaussian components, $N(x|\mu_k, \Sigma_k)$ is the normal distribution with mean μ_k and variance Σ_k and π_k the mixing coefficient. π_k should satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

To explain the steps that lead towards the computation of $P(c|x)$, Awwad Shiekh Hasan et al. introduce a random K-dimensional binary variable z_k. z_k satisfies $\in \{0,1\}$ and $\sum_K z_k = 1$. Which basically helps us compute the chance that x is generated by a certain component z_k, $P(z_k|x)$. In this case

$$P(z_k = 1|x) = \frac{\pi_k P(x|z_k = 1)}{\sum_{k=1}^{K} P(x|z_k = 1)}$$

Where $\pi_k P(x|z_k = 1)$ can be calculated from the component $z_k{}'$s multivariate normal probability density function given its mean and variance. This is also used to assign each datapoint in the train data to the component it was most likely to be generated from.

Furthermore the chance that $z_k$belongs to class i, $P(c = i|z_k)$ can be calculated as follows:

$$P(c = i|z_k) = \frac{N_{ki}}{N_i}$$

where N_{ki} is the number of datapoints in the train data that are classified as i and have been generated by z_k and N_i is the total number of datapoints in the train data that were classified as i. Again the class labels from step 3 were used however this time only 'movement' and 'baseline' were used instead of

applying the subclasses of 'movement'. In order to compute N_{ki} and N_i class labels are needed which is not allowed in unsupervised learning techniques. Therefore this method is considered to be 'weakly' supervised.

Having computed the probabilities that x was generated by component z_k, $P(z_k = 1|x)$ and z_k belonging to class i, $P(c = i|z_k)$, the probability that x belongs to class i can be computed as follows:

$$P(c = i|x) = \sum_{k=1}^{K} P(c = i|z_k) * P(z_k = 1|x)$$

When these probabilities are computed for both classes the test data is assigned the class with the highest probability when a certainty threshold α is reached:

$$|P(c = move|x) - P(c = baseline|x)| > \alpha$$

It is unclear how exactly Awwad Shiekh Hasan et al. determined the certainty threshold α using the cross validation scheme. In this replication this was solved by taking the median certainty – its standard deviation. If a datapoint could not be assigned to a class with a certainty that met the threshold, it was assigned to the class of its predecessor.

6.Onset Detection

As each window in feature space now is classified as 'baseline' or 'movement' the final decisive mechanism of onset detection can be applied. Awwad Shiekh Hasan et al. did this by looking at the classes of 11 classified windows. This onset window moves window by window over the evaluation data. When 4 consecutive windows labelled with 'baseline' are found followed by 4 consecutive windows of the 'movement' class an onset is detected. A visualisation of this is given in figure 3. Awwad Shiekh Hasan et al. use a debounce window to stabilize the signal and prevent it from 'bouncing' between activated and not activated. however as for this research only the True Positive and False Positive detections are used for evaluation this debounce window was not necessary.

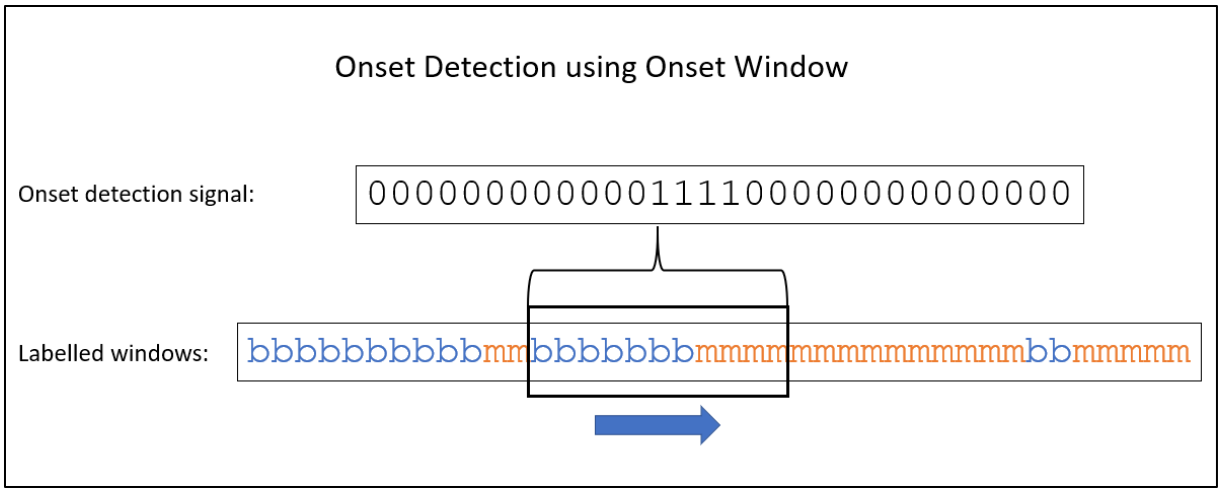

Figure 3: The onset window moving over classified windows in feature space

Evaluation

As explained the evaluation was done by using 10-fold cross validation. Around 5 to 7 trials per subject were used as evaluation data to produce the number of True Positive(TP) and False Positive(FP) detections. From this the true-false difference (TF) was computed with the following formula:

$$TF = \left(\frac{TF}{E} - \frac{FP}{E + FP}\right) * 100$$

Where E is the total number of events. A detected onset is labelled as TP if there was a real onset either 2 seconds before or after the predicted onset. This method was used by Awwad Shiekh Hasan et al. and was defined for practical asynchronous BCI evaluation(Townsend, Graimann, & Pfurtscheller, 2004).

Results

Beneath, in table 1, the results of the movement onset detection methods proposed by Awwad Shiekh Hasan et al. applied on the dataset collected by Verbaarschot et al. are presented. Besides the TF value the total number of True Positives(TP) and False Positives(FP), the average classification performance and the performance of the best model are reported. Furthermore the optimal number of components(K) and the optimal number of features(N) are also given.

Subject	TF	TP	FP	Average classification performance(%)	Best classification performance(%)	K	N
1	40.8622	37	13	57.7083	69.5833	4	10
2	46.4524	36	11	59.1583	65.5000	5	2
3	50.7463	59	18	60.4062	70.6250	6	2
4	39.0476	26	3	54.9750	66.0000	4	10
5	48.0357	38	6	65.6786	75.4167	2	6
6	66.1706	48	6	65.6250	71.6667	2	2
7	36.5000	17	3	57.6458	74.3750	3	6
8	29.6984	32	16	56.2750	69.1667	4	4
9	27.0000	14	3	63.1875	75.8333	2	6
10	26.4545	38	20	68.9866	75.6250	2	8
11	35.6652	34	14	60.5833	66.2500	6	10
12	48.9286	36	8	54.1167	68.0000	4	10
13	18.6071	17	5	62.9405	68.7500	3	8
14	44.4365	33	11	55.1000	69.0000	6	6
15	41.7489	45	16	61.9420	68.9286	3	4
16	38.7262	26	13	66.6042	78.7500	2	4
17	10.5952	14	9	60.1583	63.5000	2	2
18	42.8135	40	14	61.2143	67.1429	3	4
19	40.1984	32	6	62.5000	67.5000	3	6
20	67.6515	67	16	64.4554	73.5714	4	10
21	19.5635	22	11	59.0000	68.7500	7	4
22	50.6893	63	24	68.1736	75.6250	6	6
23	41.5833	40	13	59.3750	67.5000	4	8
24	10.5758	18	8	61.9583	71.9444	4	8
25	31.8539	44	15	56.0139	59.7222	3	10
26	48.4762	30	6	54.1500	71.2500	4	8
27	13.8093	32	31	50.5000	70.0000	3	8
28	58.8690	37	12	64.3625	72.5000	7	2
29	32.5714	23	8	55.4250	69.5000	4	4
30	17.1574	41	38	63.4271	79.0625	3	4
31	38.3492	37	16	65.0333	74.5833	8	4

Table 1: 'Weakly' supervised GMM movement onset detection results

With an average TF score of 37.5431 (std. 14.7984) the investigated method performs rather poorly. Although the classification accuracy of the best performing model is greater than 70% with roughly half the participants the TF score unfortunately does not seem to follow. This might be due to the fact that often a movement onset was not detected at all because of an absence of four consecutive windows classified as 'baseline' followed by four consecutive windows classified as 'move'. In the ideal world the classifier would start with classifying windows as 'baseline' and then a sudden switch to 'movement' would happen. This, however, is unfortunately not the case. As it turns out this process stays very gradual and the data remains lowly separable. As for the performance of the GMM, as expected it performed better with a rather low number of features and components. On average 6 and 4 respectively.

To find out whether Awwad Shiekh Hasan et al.'s methods is an improvement relative to Verbaarschot et al.'s linear classifier their results had to be converted to True Positives and False Positives. Verbaarschot et al. evaluate their results by dividing them into 4 classes according to the time of the prediction relative to the button press of the participant. According to them a predicition is 'too early' if it happens 2 seconds or longer before the button press, 'early' when it happens between 2 seconds and 1 second before the button press, 'on time' when it happens within 1 second before the button press and 'too late' if it happens after the button press. As Verbaarschot provided the exact prediction times these results could be converted to Awwad Shiekh Hasan et al.'s ranges of TPs and FPs. In the figure below a timeline of how the predictions are labelled for a single trial by the two authors is given.

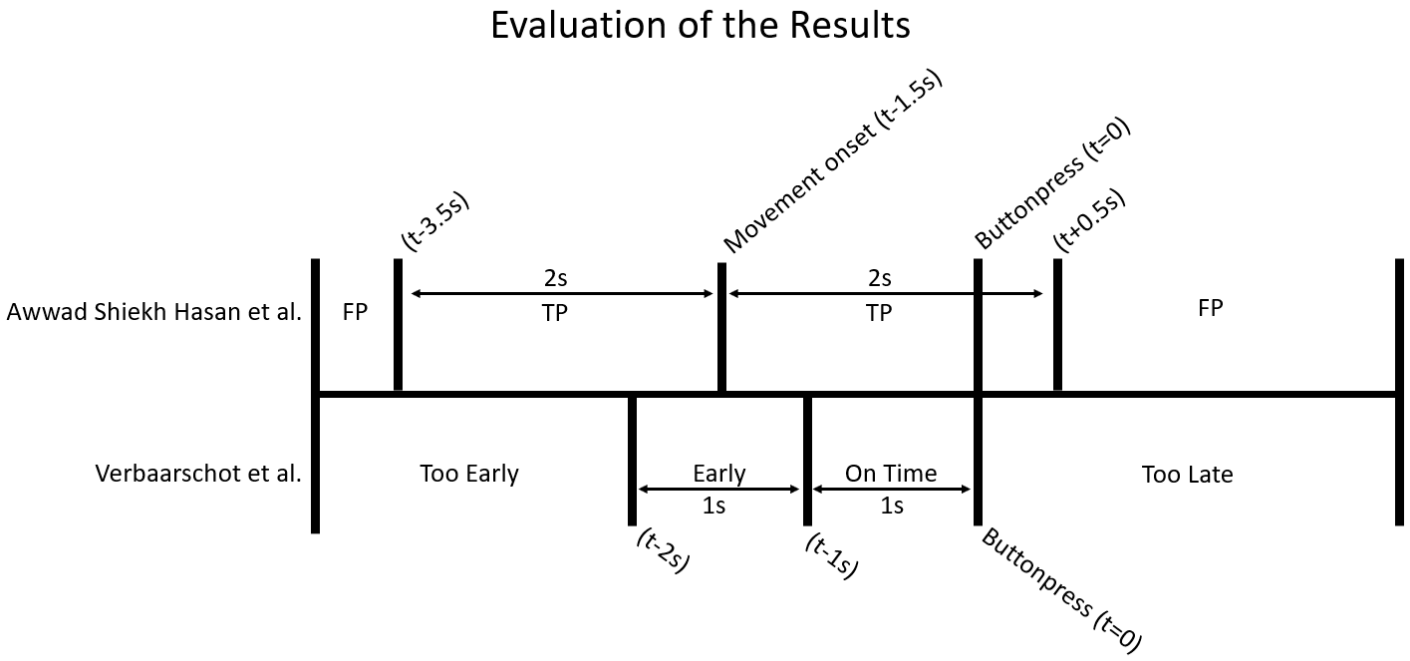

Figure 4: Evaluation of the results by both authors on a trial timeline

After converting Verbaarschot et al.'s results the TF score was computed and is given in table 2, below. Furthermore the difference in TF score is given in the last column. Here a positive score means the improvement of Awwad Shiekh Hasan et al.'s method relative to the linear classifier used by Verbaarschot et al.

Subject	TF	TP	FP	TF difference
1	-2.9126	0	3	43.7748
2	8.0000	8	0	38.4524
3	9.0099	10	1	41.7364
4	8.0392	10	2	31.0084
5	16.0099	17	1	32.0258
6	10.9295	13	2	55.2411
7	8.0874	11	3	28.4126
8	22.2314	29	7	7.4670
9	13.0222	17	4	13.9778
10	21.8206	24	2	4.6339
11	0	0	0	35.6652
12	14.5244	23	10	34.4042
13	1.0202	2	1	17.5869
14	11.0392	13	2	33.3973
15	26.0392	28	2	15.7097
16	3.7431	12	9	34.9831
17	0.7407	16	21	9.8545
18	4.2935	10	6	38.5200
19	3.0303	3	0	37.1681
20	0	0	0	67.6515
21	27.0000	27	0	-7.4365
22	25.0392	27	2	25.6501
23	4.1973	8	4	37.3860
24	26.0392	28	2	-15.4635
25	0	0	0	31.8539
26	17.8602	20	2	30.6160
27	7.0099	8	1	6.7994
28	-0.5432	6	7	59.4122
29	16.0000	16	0	16.5714
30	29.0099	30	1	-11.8525
31	8.5021	16	8	29.8471

Table 2: linear classifer movement onset detection results converted to TPs, FPs, TF scores and the improvement of TF score per subject

Finally the TF scores of both methods were tested on whether it is normally distributed using a one-sample Kolmogorov-Smirnov test. Both results were negative which means both scores are not normally distributed. As a consequence the results can not be tested with statistical methods which assume normally distributed data and thus a right-sided Wilcoxon signed rank test was done to test whether the TF scores of the 'weakly' supervised GMM performed significantly better than the TF scores of the linear classifier.

As it turns out the 'weakly' supervised GMM significantly($P<.001$)outperformed the linear classifier with the mean of the difference in TF scores being 26.614 and a standard deviation of 19.4687. Where the average TF score of the linear classifier is 10.9285 with a standard deviation of 9.5586.

Discussion & Conclusion

Although the 'weakly' supervised GMM clearly outperformed the linear classifier both methods did not perform very well. The results do not come near the TF scores (mean 80.99, std. 16.30) Awwad Shiekh Hasan et al. report. This might be due to the fact that the experiment by Verbaarschot et al. was done in a more noise rich environment where the participants could have been excited and distracted causing the EEG data to be lowly separable. Another cause for this discrepancy in performance might

be due to the fact that Awwad Shiekh Hasan et al. were only able to test 5 subjects. This might lead to a less representative TF score than might be true for the whole population.

Subject	TF	TP	FP	Best Classification Performance(%)	K	N
1	97.56	40	1	90.43	8	5
2	87.74	37	2	87.70	8	5
3	64.07	36	14	76.51	8	22
4	92.56	38	1	94.64	8	19
5	63.02	28	3	80.58	10	12

Table 3: Results as reported by Awwad Shiekh Hasan et al.

Furthermore, Awwad Shiekh Hasan et al. take a rather large window to label predictions as True Positive (2 seconds before and after the true onset). In some cases this even results in predictions after the button press being labelled as a True Positive. As reported by Libet et al. a movement onset happens up to 2 seconds before the actual movement(Libet et al., 1983). Therefore it does not seem appropriate to classify a movement onset prediction that happens either before or after this window as True Positive. By taking a window that is twice as large, a larger number of TPs can be obtained without the classifier accurately detecting the onset. When a 2 second window was applied to the evaluation method rather than the 4 second window Awwad et al. used, the new TF score was found to be considerably lower(mean 6.6951, std. 22.2024) with a lot of scores being negative. This indicates that often there were more False Positives detected than True Positives.

In conclusion, movement onset detection remains hard. As this preparatory phase in the brain is a gradual process the data seems to stay lowly separable making it hard to determine a switch from the baseline state towards movement. The methods by Awwad Shiekh Hasan et al. did improve movement onset detection in the game ‘Flip-that-Bucket’ by Verbaarschot et al. however the ‘weakly’ supervised GMM did not perform very well. Further experiments to determine how well these methods work need to be carried out in more sterile conditions to examine whether it was just the noise rich, ecologically valid, environment of the InScience festival that caused the GMM to perform poorly or that there might also be some cause in the methods itself.

Recent studies have shown that it is possible to achieve good results on movement onset detection using a variety of methods including a random forest classifier(Liu et al., 2018), a nonlinear dynamic multiple-input/single output model(Mirzaee & Moghimi, 2019), linear discriminant analysis(Zhang, Chen, Jianbin, & Meng, 2018) and neural networks(Gatti, Atum, Schiaffino, Jochumsen, & Manresa, 2019). If the goal is to develop asynchronous BCI systems for movement onset detection that have to work in everyday life it remains important to test those various methods in ecologically valid experiments to ensure they can deal with the noise that comes with it.

Bibliography

Ang, K. K., & Guan, C. (2013). Brain-computer interface in stroke rehabilitation. *Journal of Computing Science and Engineering*, *7*(2), 139–146. https://doi.org/10.5626/JCSE.2013.7.2.139

Awwad Shiekh Hasan, B., & Gan, J. Q. (2010). Unsupervised movement onset detection from EEG recorded during self-paced real hand movement. *Medical and Biological Engineering and Computing*, *48*(3), 245–253. https://doi.org/10.1007/s11517-009-0550-0

Bai, O., Rathi, V., Lin, P., Huang, D., Battapady, H., Fei, D.-Y., … Hallett, M. (2011). Prediction of human voluntary movement before it occurs. *Clinical Neurophysiology*, *122*(2), 364–372. https://doi.org/10.1016/J.CLINPH.2010.07.010

Coghlan, K. (2006). EMGONOFF - File Exchange - MATLAB Central. Mathworks.com. Retrieved from https://nl.mathworks.com/matlabcentral/fileexchange/11049-emgonoff

Gatti, R., Atum, Y., Schiaffino, L., Jochumsen, M., & Manresa, J. B. (2019). Convolutional Neural Networks Improve the Prediction of Hand Movement Speed and Force from Single-trial EEG. *BioRxiv*, 492660. https://doi.org/10.1101/492660

Hodges, P. W., & Bui, B. H. (1996). A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography. *Electroencephalography and Clinical Neurophysiology - Electromyography and Motor Control*, *101*(6), 511–519. https://doi.org/10.1016/S0921-884X(96)95190-5

Lew, E., Chavarriaga, R., Silvoni, S., & Millán, J. del R. (2012). Detection of self-paced reaching movement intention from EEG signals. *Frontiers in Neuroengineering*, *5*, 13. https://doi.org/10.3389/fneng.2012.00013

Libet, B., Gleason, C. A., Wright, E. W., & Pearl, D. K. (1983). Time of conscious intention to act in relation to onset of cerebral activity (readiness-potential). The unconscious initiation of a freely voluntary act. *Brain : A Journal of Neurology*, *106 (Pt 3)*, 623–642. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/6640273

Liu, D., Chen, W., Lee, K., Chavarriaga, R., Iwane, F., Bouri, M., … Millan, J. del R. (2018). EEG-Based Lower-Limb Movement Onset Decoding: Continuous Classification and Asynchronous Detection. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(8), 1626–1635. https://doi.org/10.1109/TNSRE.2018.2855053

Mirzaee, M. S., & Moghimi, S. (2019). Detection of reaching intention using EEG signals and nonlinear dynamic system identification. *Computer Methods and Programs in Biomedicine*, *175*, 151–161. https://doi.org/10.1016/J.CMPB.2019.04.023

Mohammadi, R., Mahloojifar, A., Chen, H., & Coyle, D. (2012). EEG Based Foot Movement Onset Detection with the Probabilistic Classification Vector Machine (pp. 356–363). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34478-7_44

Müller-Putz, G. R., Scherer, R., Pfurtscheller, G., & Rupp, R. (2005). EEG-based neuroprosthesis control: A step towards clinical practice. *Neuroscience Letters*, *382*(1–2), 169–174. https://doi.org/10.1016/j.neulet.2005.03.021

Pfurtscheller, G., & Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, *110*(11), 1842–1857. https://doi.org/10.1016/S1388-2457(99)00141-8

Repovš, G. (2010). Dealing with Noise in EEG Recording and Data Analysis. *EP Europace*.

Thomson, D. J. (1982). Spectrum Estimation and Harmonic Analysis. *Proceedings of the IEEE*, *70*(9), 1055–1096. https://doi.org/10.1109/PROC.1982.12433

Townsend, G., Graimann, B., & Pfurtscheller, G. (2004). Continuous EEG Classification During Motor Imagery—Simulation of an Asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *12*(2), 258–265. https://doi.org/10.1109/TNSRE.2004.827220

Verbaarschot, C. S., Gerrits, A. B. W., Haselager, W. F. G., & Farquhar, J. D. R. (2019). A FUN EEG-BCI GAME ON GOOEY MOVEMENT INTENTIONS.

Zhang, J., Chen, W., Jianbin, Z., & Meng, F. (2018). Detection of self-paced reaching movement intention from EEG signals. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1820–1824). IEEE. https://doi.org/10.1109/ICIEA.2018.8398004

Zhao, Y., Shrivastava, A. K., & Tsui, K. L. (2018, October). Regularized Gaussian Mixture Model for High-Dimensional Clustering. *IEEE Transactions on Cybernetics*, pp. 3677–3688. https://doi.org/10.1109/TCYB.2018.2846404