T2 Movement Prediction Based On EEG Signals with HMMs

Abstract

EEG signals are a useful tool for classifying various brain activities. Former studies have been conducted to use EEG data to classify subjects into physical states such as "movement" or "rest" and have been very successful at doing so. Such classification is applicable to various technologies such as artificial limbs and alternative human-computer interfaces. Responsiveness is a vital aspect of these technologies, and thus there is a need for more than just classification but for prediction. In this study we explore the use of Hidden Markov Models in combination with Electroencephalography (EEG) data to produce a system capable of performing prediction of limb movement.

1 Introduction

Brain-Computer Interface (BCI) systems are emerging technologies which infer commands from a users brain wave, by placing electrodes on key points of the head. As a non-intrusive method of human-computer interaction, they appeal to persons that may suffer from certain disabilities. In particular, they offer a machine control mechanism which does not requires the use of limbs or fine motor skills.

The inspiration for this problem is the work done by Gudino-Mendoza and Antelis [4] as they come to the conclusion that movement intention can be detected up to 1.5 seconds before movement execution.

Electroencephalograms are a key and fundamental tool in today's medicine for performing various disease diagnoses and real-time monitoring of patient's heart conditions. These signals can also be used to map various brain activities while performing physical or mental tasks.

1.1 Hidden Markov Model (HMM)

A Hidden Markov Model is a graphical model which is used to model phenomena in which different states of the model will produce different outcome distributions, but the state is hidden from the observer. For example consider the idiomatic casino example, in which there are two visually identical dice, one of which is biased, and the other is not. Given a series of dice rolls, the observer must guess which dice was used to make these rolls. For each iteration there is a chance that the roller has transitioned to the other die. In this case, the dice states are hidden from the observer and the dice rolls are the observable values. Natural questions arise such as can the observer conclude how many states there are? Can the observer compute the most likely sequence of hidden states based on his observations?

The motivation behind choosing to use a graphical model such as a HMM over more traditional Machine Learning techniques is the temporal information utilized by the transition matrix. Consider classifying a single time-step of a subject as rest or movement based on their EEG output, this is certainly possible, but consider that now you are also aware that in the subjects previous time-step they were in rest. They are now much more likely to be in rest at this moment as well. Indeed the transition matrix of an HMM translates nicely to the transitional nature of physical movement.

There are three basic problems in regards to HMMs [1] (i) Given a series of observations and a specified HMM, compute the probability of observing that sequence given the parameters of your HMM. (ii) Given a series of observations and an HMM, compute the highest likelihood state sequence that will have generated the observations. (iii) Given a series of observations, adjust the parameters of your model to better fit the observations (Learning).

In experiment 1 we train different HMM's for rest, intention, left-arm and right arm movement and use (i) to choose the HMM which most likely produce the given observations. In experiment 2 we train a single HMM and use (ii) to classify the most likely state sequence given the observations, where the states are explicitly defined as rest, movement, and intention. In both cases we are using (iii) to generate out models from the data.

1.2 The Task

Current work has confirmed the ability to identify, using EEG signals, when a subject transitions from rest to movement[4]. Our task is to further this technology by introducing a third state, intention, that directly precedes movement, and build a model that can accurately identify this state. We do this by introducing an artificial label in our trial data, that is the intention label. Time-steps preceding the initiation of movement by a fixed intention length window are declared as intention.

For experiment one the learning task is to produce multiple models, one for each physical state. For example we wish to generate a rest HMM which represents the transitional and observational tendencies of rest, and a similar model for intention, left and right arm movement. Then, classification becomes selecting the model which produces the highest likelihood of generating the observations at a segment of each trial. Thus the performance task is classifying segments in the signal correctly.

For experiment two the learning task is to produce a single HMM with precisely four hidden states, one for rest, intention, left and right arm movement. Then, given a trial, the HMM will output the most likely sequence of hidden states which would produce such observations. In this case we are evaluating accuracy over the entire trial, considering for every window whether the HMM has correctly classified the hidden state.

1.3 Related Work

In similar work, significant event-related de-synchronization was found in the motor-related alpha and beta frequency bands in the moments preceding movement[4]. Implies were that intention can be detected approximately 1.5s before movement execution onset. This result could be used in real time to trigger an assistant device for active motor rehabilitation therapy. Intention was detected in 78% of trials. This work was done using an SVM classification.

In relevant studies [6] the majority of tasks have been experimenting with methods for classifications and extraction of Motor Imagery. Motor Imagery corresponds to the mental state of an individual while performing an action. Feature selection for classification of EEG signals can be achieved with strong results. In this experiment we go beyond classifications to create a system capable of predicting movement intention of a person based on signal readings from a EEG input.

2 The Study

Both experiments we conducted are based on the same data source from a single study and a set of open source tools for Matlab.

2.1 Observations

The data we used is the same data-set used in [4]. It consists of 18 subjects with 96 trials each. A single trial consists of a subject sitting comfortably on a chair with their arms resting. A screen was placed in front of them and offers visual cues to guide them through the experi-

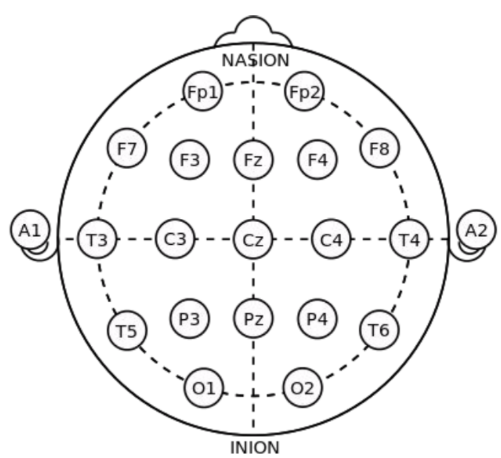


Figure 1: EEG Electrode Placement

ment. The first cue showed the text 'relax' for three seconds, and participants were asked to not imagine or execute any movement at this time. The second cue showed an arrow pointing either left or right. The second cue lasts 12 seconds and participants were asked to move the corresponding arm towards the middle of the screen, not immediately, but any time they feel like it after 5 seconds without counting in their head. Immediately after moving, participants return their arm to the chair and a 'rest' cue is shown on the screen (See Figure 2). Participants were all able-bodied right-handed subjects without diagnosis of neurological nor motor disease.

2.2 The Data

During the trials, EEG signals were recorded using 21 electrodes, positioned according to the 10/10 international electrode location system. Trials were trimmed from the presentation of the first cue to the presentation of the third cue. The timeline of a trial is then remapped such that movement begins at time $t = 0$ and the intention phase is 1.5 seconds preceding movement. That is the intention phase consists of time $t \in [-1.5, 0]$ and the rest phase consists of time $t < -1.5$.(see figure 3).

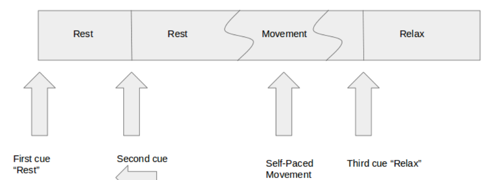


Figure 2: Initial Trial Data

After the performed pre-processing, the data we are working with is 18 subjects, 96 trials each and 9 electrodes of interest, that is F3, F4, Fz, C3, C4, Cz, P3, P4 and Pz. This results in 9 input signals for each trial, sampled at 256 Hz.

2.3 The Tools

For the programming segments of the experiments we resourced an open source library PMTK3[9] for use with Matlab to perform our tasks.

The library has a vast set of tools including decent support for latent variable models such as Hidden Markov Networks. For HMMs specifically inference was performed by using the forward-backward algorithm to compute posterior marginals of all hidden state variables.

For sequence prediction, the Viterbi algorithm was used to predict the most likely sequence of hidden states given a set of observed data.

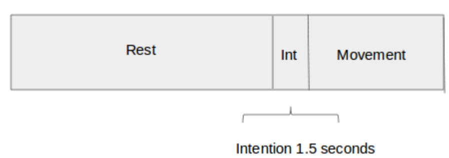


Figure 3: Cropped And Labeled Data

3 Experiment One - Multiple HMM's

3.1 The Motivation

In experiment one we utilize an HMM's ability to give the probability of producing a particular sequence of observations given the model parameters. The idea is to train a separate HMM for each segment of trials i.e. rest, intention, left-arm, right-arm and one for general movement. Then, in classification, given a series of observations with hidden labels, we can compute the likelihood for each HMM to produce such a sequence and choose the model which produces the highest likelihood. The movement HMM is not used in our general classification, but for a separate statistic, evaluating its performance solely against the rest HMM.

To be more precise, suppose we are testing the accuracy of our models on a test trial. We take a single window of observations and compute the likelihood of that sequence being produced by the rest, intention, left-arm and right-arm HMMs. If the window is from the rest phase of a trial, and the likelihood produced by the rest HMM is the highest, then our models have produced a correct classification. In the case of the movement HMM, during a movement phase, we take the log-likelihood of both the movement and rest HMM, and in the case that the movement HMM is more likely, we classify that as a success. Therefore, the movement HMM is only involved in evaluating false negatives.

3.2 Windowing

Classification of a sequence of observations is done by comparing the likelihoods for each model to produce such a sequence. This raises the question, what is the optimal observation sequence length? At one extreme, we have the entire trial as the window, which incorrectly groups all states between which we are trying to discriminate. At the other extreme, classifying a single observation at a single time-step does not make sense as it disregards all transitional information. Hence, the optimal sequence length is somewhere within the range of $[1, n]$ where n is the number of observations in a given trial. We refer to this length as the window size. An optimal window size is not known and is thus another hyperparameter in our experiment that we investigate over a range. The range is softly related to the sample frequency (256 Hz), and the brain frequencies of interest (5-40), as a window must be long enough to sample multiple periods of any of these frequencies. Another consideration in choosing a window size that is the length of the defined intention window before movement, as the intention segment should consist of multiple windows.

3.3 Hidden States

Since we are using a separate HMM for each physical state, we are making the assumption that there are hidden states within each of them. It is possible, as an example, that left arm movement is composed of 3 phases, *Acceleration, Stop, and Deceleration*.These states are not observed, nor do we have labels for them and as such, we vary the number of hidden states as a parameter in our experiment to optimize our results.

3.4 Process

3.4.1 HMM Training

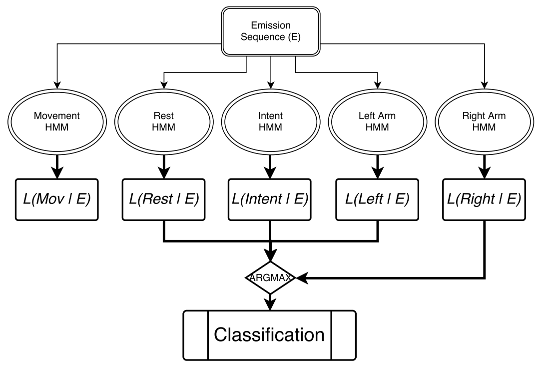
EEG signals are unique to each person, therefore we are unable to train one HMM for each patient using all of their data. Thus for each subject, a new experiment is performed. For each subject we have 96 trials, and randomly split the data into 80% (76) training data and 20% (20) test data. For each trial in the training data, we split it further into rest, intent, left-arm, right-arm, and movement (both left and right arm) blocks. Thus we were left with 76 sequences of movement, rest, intent, and 36 sequences of left-arm and right-arm movement for training. The test data remains intact as 20 trials with rest, intent, and single arm movement.

In order to train each HMM, we feed it the raw EEG training sequences to which it corresponds. For example, the left-arm HMM received the 36 sequences of left arm movement, and the movement HMM received both the left-arm and right arm sequences. The HMMs were trained using PMTK3's *hmmFit* function which takes in all of the training trials, the number of states (which we varied) and the type of HMM, in our case Gaussian, and applies the training data to the *Expectation Maximization* algorithm to best fit the data.

3.4.2 Evaluation

As mentioned, since each person's brain signal is unique, we performed the experiment for each patient. We grouped the results by taking the average. That is, for each of the 18 patients we trained separate HMMs, recorded the accuracy for each HMM and took the average. Every patient had 20 test trials, which we split up into rest, intent and movement sections. We classified rest sections as any pre-movement sequence, intent sections as the windows before movement within the intention length parameter, and left or right arm movement based on the labels. At each window, we take the log-likelihood from the *Rest*, *Intent*, *Left-Arm* and *Right-Arm HMM* using PMTK3's *hmmLogProb* function, and our classification is the HMM with the highest likelihood (See Figure 4). Therefore at chance, we would expect 25% accuracy and in the case of the *Movement HMM*, since we only measured false negatives at chance we would expect 50% accuracy.

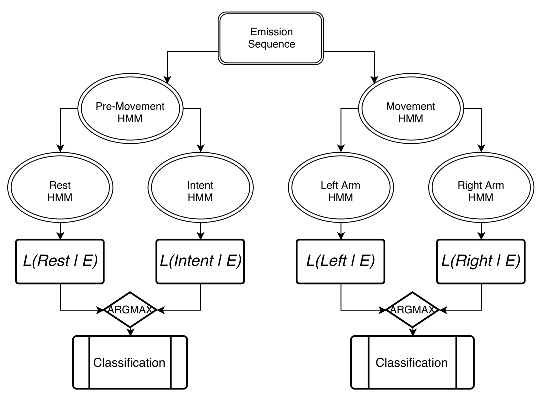
Figure 4: Five Separate HMMs



3.4.3 Modification

Using the same training methodology, we tried to put the HMMs in a hierarchy (See Figure 5). Under the assumption that the biggest difference in brain signals is between pre-movement and movement, we used the rest HMM and movement HMM to classify a signal. If the signal was deemed to be rest, we compared rest against intent and made taking the *argmax(intent, rest)* as the classification. On the other hand, if the signal was deemed to be movement, our classification was the *argmax(left-arm, right-arm)*.

Figure 5: HMM Hierarchy



3.5 Results

3.5.1 Iteration One

In the original iteration of the experiment, we varied the number of hidden states n, intent length L and window size w within

$$n \in \{2, 3, 4\}, L \in \{32, 64, 128\}, w \in \{16, 32, 64, 128\}$$

with the added constraint that window size must be smaller than intent size. In this case, the window size and intention length are in the number of time-steps, sampled at $256Hz$, thus an intention length of 64 is equivalent to $\frac{64}{256} = \frac{1}{4} seconds$. The results are shown below as the highest average accuracy, found with $n = 3, L = 128, w = 64$.

Rest	Intent	Left Arm	Right Arm	Movement
66.4%	47.28 %	37.05 %	81.23 %	52.15 %

We believe that the most striking result of our experiment is the difference between right arm and left arm accuracy. Though we are unsure of the true cause, it is possible that the handedness of the participant plays an important role in this outcome - especially since all of our participants are right-handed. It is important to reiterate that our assignment of left and right arm is arbitrary and could potentially be that left-arm accuracy was 81.23 %.

In this case, the movement accuracy is solely compared against rest, and thus with an accuracy of 52%, it is at the chance level. It is most likely that our model did not capture the proper distinctions between movement and rest, and it would be wise to further investigate this before claiming that brain signals from individual arm movements differ drastically. Because 52 % accuracy was not the best accuracy on movement we achieved, we thought it would still be worthwhile running the modified experiment to see if we would achieve better results.

On the other hand, the rest result of 66.4% is significantly above the chance level. It is possible that because the rest section had by far the most data, the Gaussian distributions had tighter curves and produced better log-likelihoods as a result. This could be verified by adapting the data-set to have equal length movement and rest sections.

3.5.2 Iteration Two

In the modified iteration of the experiment, keeping the same restrictions and format as before, we fixed hidden states n, intent length L and window size w within

$$n = 3, L \in \{64, 128, 256, 512\}, w \in \{64, 128\}$$

We found that the best average results were found with $n = 3, L = 512, w = 64$.

Rest	Intent	Left Arm	Right Arm	Movement
70.74%	22.53 %	32.43 %	63.65 %	50.29 %

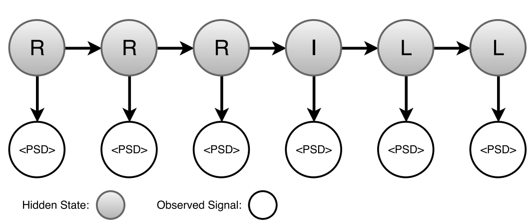
We see a lot of the same patterns here, such as approximately two thirds accuracy for rest, approximately double accuracy for right-arm over left arm, and chance accuracy for movement. The important distinction is that intention accuracy significantly drops, and below chance. It would then follow that intention is either not measurable in this method, or that intention waves are very different from rest waves, and should not be classified within the pre-movement hierarchy. With the movement HMM inability to classify movement data, we believe that under our current model, movement patterns should not be grouped together, and have significant enough differences to warrant separate HMMs or new models.

4 Experiment 2

4.1 The Motivation

In Experiment two we utilize the Viterbi algorithm for determining the most likely sequence of hidden states given a sequence of windows. Unlike experiment one, we construct a single HMM for which we define the number of hidden states to precisely match the number of physical states that we are classifying, which are, rest, intention, left and right arm movement. The idea is that states are more homogeneous than experiment one assumed, and it would be better to use the temporal information of the state transitions to analyze a trial.

Figure 6: HMM Sequence



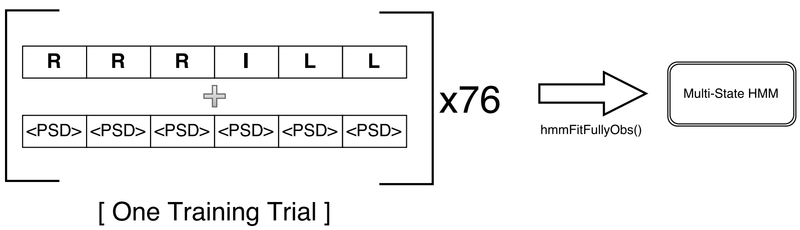
4.2 FFT/PSD/Data Processing

In experiment two we transform data from the time domain to the frequency domain using Fast Fourier Transform. We take the approximately 1700 time steps, each of which has 9 values of the observed brain signals, and split them into windows like we did in experiment one. We take each window and use a *pwelch* transformation to get the component waves at each of the 9 electrodes, in the range of 6-40Hz, giving us 315 frequency values for each window to use as a feature vector.

4.3 Learning

Learning an HMM is the process of adjusting the model parameters to better the represent observation sequences. This raises an interesting observation in the format of our data. Since each trial is guaranteed to begin in the rest state, we can fix the initial distribution vector to match this property. Similarly, the state transitions are known to a certain extent, since in a trial we know that states can only either transition to themselves or transition from rest to intention, and intention to movement. We can leverage this prior knowledge in order to give a greater foundation for our model to learn an accurate emission matrix. To learn the HMM, we give PMTK3's *hmmFitFullyObs* a sequence of full trials, along with labels, which returns a trained HMM (See Figure 7).

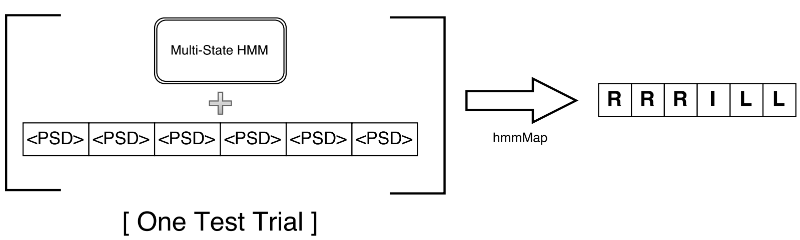
Figure 7: Train Fully Observed HMM



4.4 Evaluation

For classification we input an entire trial of observations, and our HMM into the Viterbi algorithm, know as *hmmMap* in PMTK3, which outputs the most likely sequence of states to match the trial (See Figure 8). For each observation, we will then know the most likely hidden state that our model will have been in at that time. Classification can then be a ratio of correct guesses to number of windows.

Figure 8: Classify Test Sequence

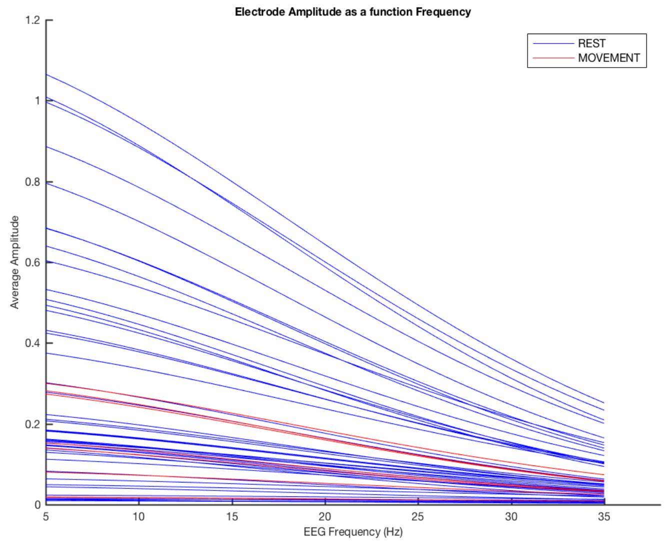


4.5 Results

We encountered a problem when we tried to use *hmmMap* to get the most likely set of states - it predicted sequences of full rest. We learned because our feature vector was so large, the log-probability of transitioning was *-Inf*, and as we started in the rest state, the HMM predicted it would remain in rest. Unfortunately, reducing the size of the feature vector by averaging the amplitudes across electrodes, did not improve the probabilities, and our results stayed the same.

Figure 9. shows that when you analyze the average the amplitudes over a window for each frequency and compare the results between rest and movement, we see a possible explanation for the behavior. Notice that the movement frequencies have enormous overlap with the rest frequencies, and thus it is possible that our HMM cannot sufficiently distinguish between the two signals.

Figure 9: Average Amplitude across Windows for Electrode 1



5 Conclusion and Future Work

In this paper we have discussed how we used Hidden Markov Models to analyze Electroencephalography data for patients moving their right and left arms. By keeping the data in the time domain and using multiple HMMs with unknown states we achieved moderate results distinguishing rest and right-arm movement. We also attempted to use a single HMM with fully observed, transformed data to capture the temporal properties of the sequences. Though conceptually it appeared to be a more effective strategy, we were unable to achieve any meaningful results which we believe was due to overlaps in the data combined with a strict transition matrix.

In the future, experiment one could be further modified by increasing the amount of movement data, and trying similar strategies with in the frequency domain rather than time. By adding more transitions between rest and movement in the trials, it is possible that our transition matrix in experiment two would have been better suited to handle subtle differences. Further, if we had more transitions it is likely that learning transition signals between phases would be better represented, and more feasible to model the intent window as a longer sequence.

Finally, our results indicating a difference between classification accuracy of left and right arms which implies the need for further examination.

References

[1] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE Proc. IEEE, 77(2), 257-286.

[2] Modeling motion patterns of dynamic objects by IOHMM. Zhan Wang, Rares Ambrus, Patric Jensfelt and John Folkesson

[3] Input-Output HMMs for Sequence Processing. Yoshua Bengio and Paolo Frasconi, Member, ZEEE

[4] Detecting the intention to move upper limbs from electroencephalographic brain signals. Berenice Gudino-Mendoza, Javier M. Antelis

[5] A review of classification algorithms for EEG-based brain–computer interfaces. F Lotte, M Congedo,ALecuyer, F Lamarche1 and B Arnaldi

[6] Evaluation of feature selection and classification methods for recognizing motor imagery tasks from electroencephalographic signals Roberto Vega · Touqir Sajed · Kory Wallace Mathewson · Kriti Khare ·Patrick M. Pilarski · Russ Greiner · Gildardo Sanchez-Ante · Javier M. Antelis

[7] Koller, Daphne, and Dan Jurafsky. "Coursera - Free Online Courses From Top Universities". Coursera. N.p., 2016. Web. 3 Mar. 2016.

[8] Aziz, Fayeem, et al. "HMM based automated wheelchair navigation using EOG traces in EEG." Journal of Neural Engineering 11.5 (2014): 056018.

[9] PMTK is a collection of Matlab/Octave functions, written by Matt Dunham, Kevin Murphy and various other people. https://github.com/probml/pmtk3/blob/master/README.md