Classification of Brain Wave (EEG) Data

Ankur Garg
agarg12@ncsu.edu

Sanket Shahane
svshahan@ncsu.edu

Abstract

The classification of Brain Wave data of a subject can be done using various classification techniques like Logistic Regression, Support Vector Machines, Linear Discriminant Analysis, Random Forest Classifier etc. But these classifiers do not capture the sequential relationship between the EEG signals of two similar time stamps. To efficiently classify EEG data, algorithms that generate models based on this sequential relationship are preferred. In this report, we analyze how methods like Hidden Markov Models, Feed Forward Neural neural network, LSTM neural network and Structured Perceptron to train HMM can be used in the context of EEG data classification. These models are based on sequential relationships between data samples and hence, are expected to perform better. We compare the results obtained with the non-sequential algorithms.

1 Background

The objective of the project is to efficiently classify an EEG signal. An EEG signal is a representation of brain waves of a person collected by using various electrodes connected to the person's skull. The ability to identify the action a person is thinking about, based on the EEG signals collected from person, is an important part of building a robust Brain Computer Interface system. The task at hand is a classification problem. There are various standard techniques that can be used to solve a classification problem, like, Support Vector Machines, Linear Discriminant Analysis or something simple like Logistic Regression.

There have been many attempts at classification of EEG signals using such techniques. It was observed that the first step in case of EEG data classification is feature extraction. This is because the EEG data is noisy. In this report, we describe the methods used in our project for feature extraction, the performance of various standard classification methods performed on our dataset and finally we try some newer techniques like Feed Forward Neural Network, LSTM Neural Network and Structured Perceptron to train Hidden Markov Models on the EEG dataset.

The problem statement of the project is part of the Brain Computer Interface III competition [1]. The dataset being used has been sourced from this competition. The results from this competition has been used as reference to evaluate the performance of the methods used.

1.1 Similar Work

Some of the papers used for understanding the use of HMM for classification of EEG data: The paper [2] presents two different approaches in classifying the EEG data. Their work involves comparison of Linear Discriminant technique and use of Hidden Markov models. We studied the approach they used in designing the hidden markov model suitable for the EEG data. A lot of focus has been kept on data collection, and preprocessing the raw signals. Their methodology is suited for online classification of the EEG data. They have shown a block diagram of the process used to convert the raw signals into a format usable by the HMM and make classifications on the fly.

Another paper read was [3]. This paper presents results of extensive list of algorithms used and compared to classify EEG data. Important information we could get from this work is that; an algorithm which performs best on one subject does not necessarily perform best on other subjects.

2 Methodology

The baseline performance for the project were the accuracy measures of the methods submitted as part of the BCI III competition [1], from which our dataset has been sourced.

For feature extraction and dimensionality reduction, Principal Component Analysis was used. For classification, various techniques like Random Forest Classifier, Linear Discriminant Analysis and Support Vector Machines were used. The performance of these classifiers was measured on both the raw data set and the principal components to validate the requirement of dimensionality reduction. The reason for choosing these techniques was that most of the submissions on the competition had used similar techniques. So, the first step as part of the project was reproduce similar results using similar techniques.

In the next step, we used Hidden Markov Models, Feed Forward Neural Network, LSTM Neural Network and Structured Perceptron to train Hidden Markov Models. One of the properties of the EEG signal data is that the signals are sequentially related. Most of the classifiers used in the competition submissions do not capture this sequential relationship between the continuous EEG signals. This is the reason that methods like Hidden Markov Models which capture such relationship are expected to perform better in such cases. This was the motivation behind using methods like LSTM Neural Network and HMM.

The details of each method used are presented in the next section.

3 Plan and Experiments

3.1 Dataset

The data is collected from 3 subjects over 4 sessions each. During each session, the subject is asked to concentrate his thoughts on a particular action for some duration and then asked to switch the action. The frequency of recording the data is 16 Hz. This collected data is then labeled according to the instructions given during each of the sessions. In total, we have data of 12 sessions for the 3 subjects combined.

In the dataset used, there are three classes. The subject is thinking of moving his left arm, thinking of moving his right arm, thinking of generating words with some random letter.

3/4 of the data is used for training and 1/4 is used for testing. In experiments where the models are being trained on individual subjects, 3 sessions are used for training and cross-validation and the 4th session is used for testing. In experiments where the models are being trained on all 3 subjects together, every 4th session for each of the subject is used for testing. Thus, 9 sessions are used for training and 3 for testing.

3.1.1 Preprocessing and Exploratory data analysis

Training samples were 10528, 10400 and 10288 for subject 1, 2 and 3 respectively. Testing samples were 3504, 3472, 3488 respectively. Scaling was performed on the data of individual subject and all together based upon the type of experiment we had to perform. The scales of attributes were different and would have affected the results of PCA and the performance of the algorithms we would use.

On analysis, it was found that the dataset did not have missing data problem or the class imbalance problem as shown in Figure 1.

3.1.2 Correlation among attributes

Very few attributes have correlation among them. Specifically, two pairs were found with high correlation. We removed on from each pair since correlated attributes affect PCA which was later used to reduce the dimensionality. Figure 2 and 3 show the correlation matrix before and after removing correlated features.

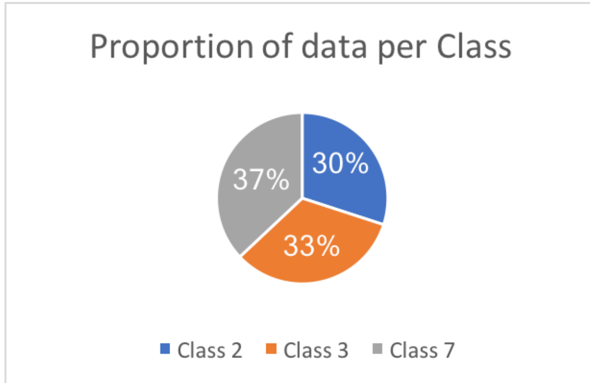

Figure 1: Class Distribution

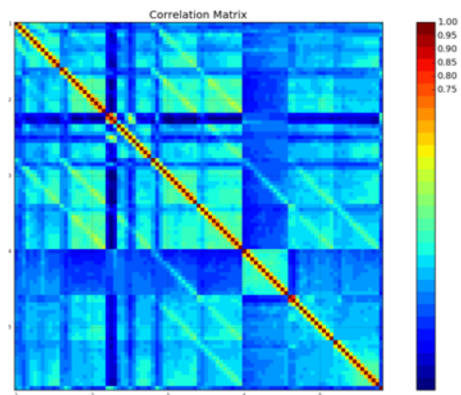

Figure 2: Before removal of Correlation

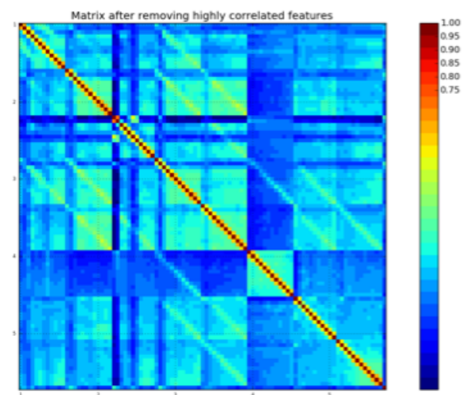

Figure 3: After removal of Correlation

3.1.3 Dimensionality

96 features collected from 32 electrodes of the EEG cap. The semantics of the features are same for all the subjects.

PCA is used to reduce the dimensions to cover about 97% variance. Since, the subjects are independent we had to fit PCA models separately on each of the subject. As a result, we required different number of components to explain the same amount of variance for different subjects. Details are as follows:

1. 40 components for Subject 1
2. 50 components for Subject 2
3. 60 components for Subject 3

For experiments involving training on all the subjects together, PCA was done on the whole training data together. As a result, 55 components were required to explain 97% variance.

3.2 Research Questions

- Does modeling the sequential relationship in the EEG data help in improving the predictions?
- Can we generalize the models learned on the data of one subject to make predictions about another subject?
  - If the answer to this question is yes, we can train our models on a limited number of subjects and generalize those models to make predictions about any random subject.

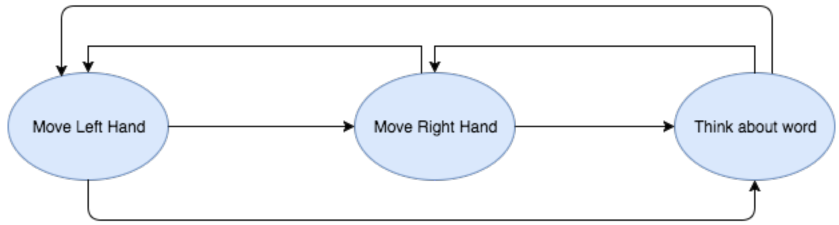

Figure 4: HMM State Diagram

- Advantage: We can help the differently-abled people to better communicate by collecting EEG data and using trained models to make the predictions which could be communicated to another subject.

3.3 Experiments

3.3.1 Hidden Markov Model

The EEG data has sequential relationship among the samples being continuously collected from a subject. HMM tries to capture that relationship as it tries to calculate a probabilistic model based on the transition probabilities between the various classes and output transmission probabilities calculated from the data. In the present case, as mentioned earlier, there were three classes in the data. This implies that the HMM representation for this data would consist of three states each representing a class (a thought action of the subject). The data provided was used to calculate the transition probabilities among these three states.

Standard HMM usually has discrete outputs for each state. In this case, the EEG data was continuous. So, the emission probabilities for the HMM were represented in the form of a multi-variate Gaussian distribution on the principal components. For predicting the class for any test sample, Viterbi algorithm was used. Figure 4 shows the representation of the various states present in the data.

3.3.2 Feed Forward Neural Network

For training a neural network a stacked vector approach was used. In this approach, for predicting a class label for the EEG data, the previous 7 samples along with the current sample were used together to predict the class label for the current sample. So, 8 samples were stacked together to form a single vector. Each sample had 40 features. So, the input layer of the network was of length 320. The reason behind this methodology was that the signals were sequentially correlated and thus, it was our impression that using the previous samples would yield better results as it would try to capture the sequential relationship between the samples. Figure 5 shows the neural network design that was used.

Specifications for the neural network used were as follows: 2 Hidden layers consisting of 12 and 8 nodes each. With an output layer of 3 nodes (each representing one of the three classes in the data). The neural network was trained with a batch size of 150 and 500 epochs.

3.3.3 LSTM Neural Network

Another approach that was used was to use an LSTM layer while designing the neural network. LSTM networks are a special type of neural networks which store/remember the previous input. This type of neural network is especially useful when you have some relationship between the data samples and you want to model it. The LSTM NN is an advancement of the Recurrent Neural Network and does not suffer from the problem of exploding gradients. Each neuron in the LSTM Layer has 4 gates inside it with each having a specific functionality. For e.g. the forget gate is the reset gate which will clear the memory of the neuron when it's output is high.

The data was transformed so that it can be given as input to such a network. The data was transformed into 3 dimensional data structure in which 8 samples (with 40 dimensions each) were together given as a single input to the network. Figure 7. shows how the data was reshaped to a 3-D mapping. The neural network design which was used to train this modified data is as shown in the Figure 8.

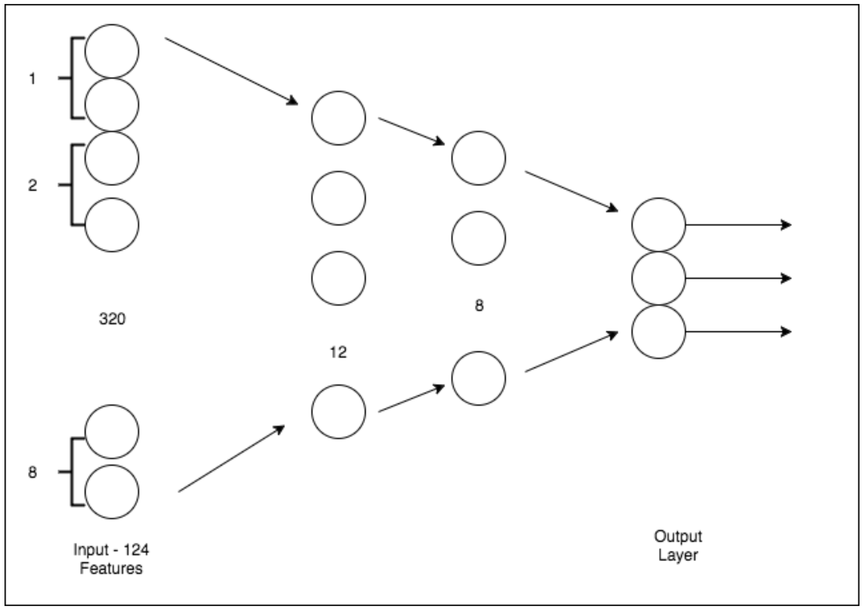

Figure 5: Feed Forward Neural Network

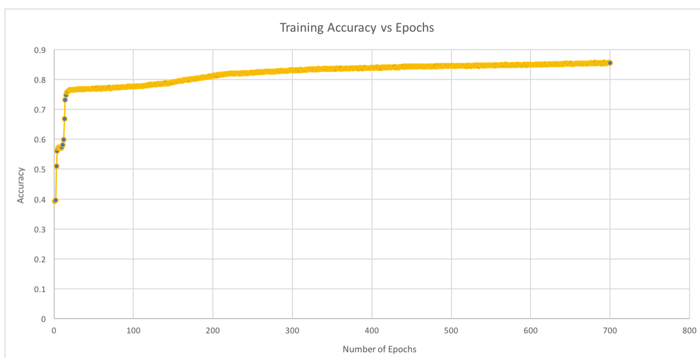

Figure 6: Neural Network Over-fitting

Here, after the input layer, a LSTM layer with 12 neurons was added, followed by a fully connected output layer, with 3 neurons (same as feed forward neural network). Finally, the output from these three neurons is converted to one-hot encoding.

3.3.4 Structured Perceptron

Structured Perceptron algorithm is used to train the Hidden Markov Model. Direct estimation of the initial and transition probabilities from the data did not prove of much benefit in our previous experiments. Structured perceptron follows a vector approach in training the model. Each data instance along with its class label is represented as a vector. The perceptron is also a vector which incrementally aligns itself in the direction which can make a distinction between two classes.

3.3.5 Post Processing

Window smoothing was used to filter out any noise in the class output labels. The reason behind this was that the data is sequentially related and a single sample with a different label than most of the samples around it would be most probably noise. It does not make sense that a subject thinking on a particular action suddenly changes to a different class for 1/16th of a second. So, window smoothing made sense.

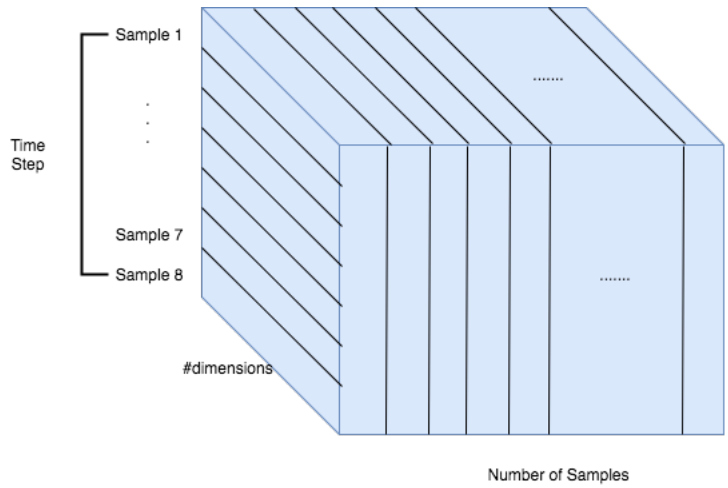

Figure 7: Data Representation for LSTM

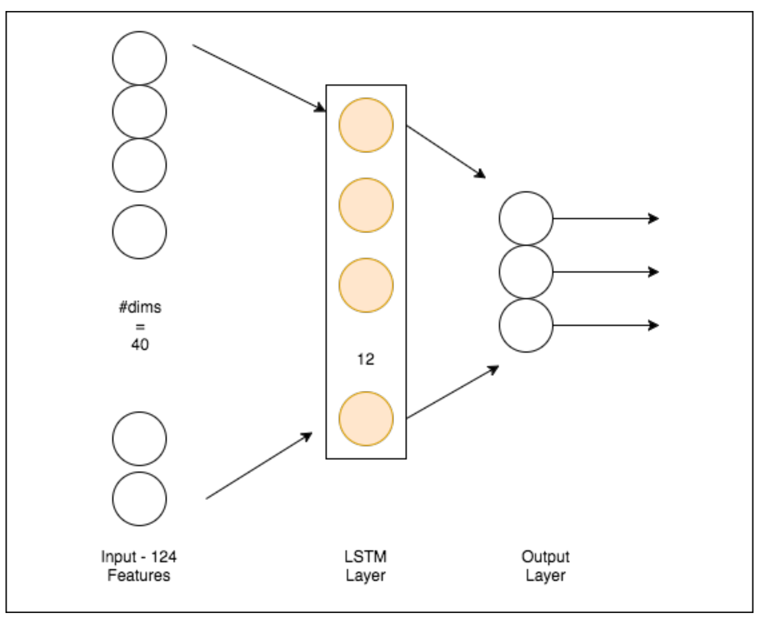

Figure 8: LSTM Neural Network - Design

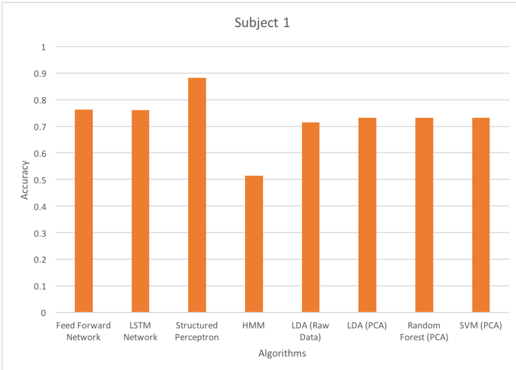

Figure 9: Results for Subject 1

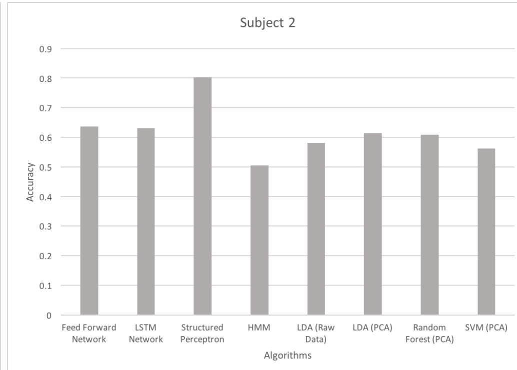

Figure 10: Results for Subject 2

4 Results

4.1 Initial Non-Sequential Classifiers

Table 1 lists the results obtained from the various baseline classifiers used.

Table 1: Test Accuracies for LDA, RF and SVM

Subject	LDA (Raw Data)	LDA (PCA)	Random Forest (PCA)	SVM (PCA)
Subject 1	71.46%	73.26%	73.14%	73.11%
Subject 2	58.12%	61.34%	60.88%	56.13%
Subject 3	49.17%	50.54%	47.70%	49.60%

4.2 Sequential Classifiers

The Table 2 lists the results obtained from the HMM (without perceptron), Feed Forward Neural Network, LSTM Neural Network and HMM (Structured Perceptron). The accuracies for the simple HMM were quite low - even lower than the non-sequential classifiers. The accuracies for Neural Networks were better than the initial non-sequential classifiers. Feed Forward and LSTM neural network results were comparable. The results obtained by using the Structured Perceptron for training HMM yielded best results out of all methods used.

Table 2: Test Accuracies for Simple HMM, Feed Forward NN, LSTM NN and Structured Perceptron

Subject	Simple HMM	Feed Forward Neural Network	LSTM Neural Network	Structured Perceptron
Subject 1	51.38%	76.2%	75.99%	88.27%
Subject 2	50.49%	63.56%	63.2%	80.21%
Subject 3	38.46%	47.21%	45.31%	57.97%

4.3 Post-Processing

Window smoothing did not yield much improvement in the results.

4.4 Evaluation of Results

The Figure 9, 10 and 11 show the comparison between the various algorithms for all three subjects. It is observed that using Structured Perceptron to train the parameters of the HMM yielded best results across all subjects. Based on the results, the answers to the research questions are as follows:

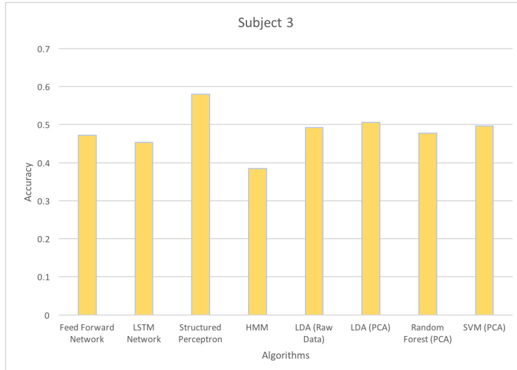

Figure 11: Results for Subject 3

- Yes, modeling the sequential relationship in the EEG data has helped in improving the predictions. This was first evident in the stacked vector approach in the feed forward neural network.
- By cross testing the models it was observed that models trained on one subject don't generalize well on other subjects. For e.g. we cross test models by training on Subject 1 and then testing it on Subject 2 and 3 Thus, we would need some training to map the signals of a new subject to the actions. However, to make any inferences statistically sound we need more data from many different subjects. Our answer to the second research question is based on extrapolation of our results.

5 Conclusion

It was observed that methods which incorporate the sequential relationship between the samples such has Hidden Markov Models (when trained with appropriate methods like Structured Perceptron) and LSTM Neural Networks perform better than the non-sequential methods like Random Forest Classifier, SVM etc. on EEG data.

Dimensionality reduction techniques like Principal Component Analysis on EEG data helps reduce noise from the data and helps extract appropriate features from the data.

LSTM networks require large amounts of data for better training. In case of absence of large amount of data, algorithms like Structured Perceptron can be used to train a Hidden Markov Model.

Also, it is observed that data from Subject 3 provided has high noise component or has some form of error in terms of labels provided. That is why, consistently every classifier performed worst on subject 3 data.

References

[1] BCI Competition III: Classification of continuous EEG signals http://www.bbci.de/competition/iii/

[2] B. Opermair, C.Guger, et al. Hidden Markov Models for online classification of single trial EEG Data. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.415.6726&rep=rep1&type=pdf

[3] Omar AlZoubi, et al. Classification of Brain Computer Interface Data. http://crpit.scem.westernsydney.edu.au/confpapers/CRPITV87AlZoubi.pdf

[4] Hidden Markov Models : A Continuous-Time Version of the Baum-Welch Algorithm. http://www.doc.ic.ac.uk/teaching/distinguished-projects/2010/m.zraiaa.pdf

[5] PCA+HMM+SVM for EEG pattern classification http://ieeexplore.ieee.org/document/1224760/?reload=true