Hand Movement Classification via EEG

Mohamad H. Alizade | 96109866

Supervisor: Dr. Hajipour

Phase I

Method

The first stage of the processing is frequency domain filtering. Different frequency bands were tested and 15-33Hz resulted in the best performance. The alpha band also showed promise but performed noticeably subpar. To keep the generalization error to a minimum, we continue with the beta band and thus achieve a simpler model. All channels were filtered out.

Since all channels are spatially correlated, we need a whitening transformation. It would've been ideal if this linear transformation also maximized the variance between the two classes. Common Spatial Pattern (CSP) subspace decomposition achieves both of the requirements. The following calculation shows why.

X_R : Right hand class data

X_L : Left hand class data

$$R_R = \mathrm{cov}(X_R) = X_R X_R^T \;,\; R_L = \mathrm{cov}(X_L) = X_L X_L^T$$

$\bar{R}_L , \bar{R}_R$ is the avg in their class

$$R = \bar{R}_R + \bar{R}_L = U \Sigma U^T$$

$$P \triangleq \Sigma^{1/2} U^T \longrightarrow S_R = P \bar{R}_R P^T ,\; S_L = P \bar{R}_L P^T$$

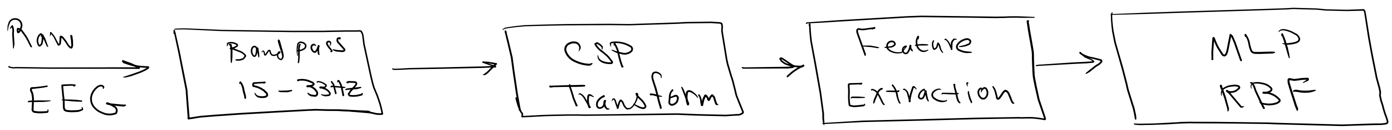
Now decomposing S_R , S_L into eign state:

$$S_R = V \Sigma_R V^T ,\; S_L = V \Sigma_L V^T \text{ and } \Sigma_R + \Sigma_L = I$$

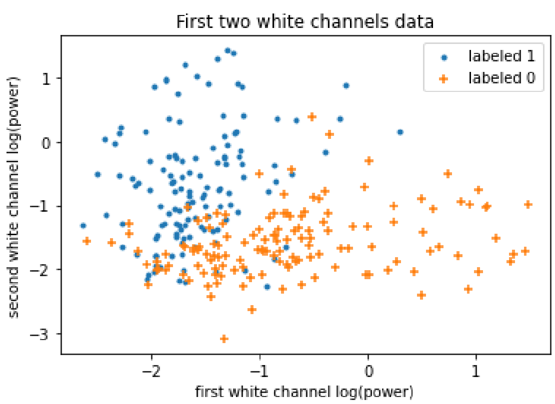
The two decompositions share the same basis. Now since the sum of the eigen values is one, the eigen vectors with the largest eigenvalues for S_R have the smallest eigenvalues for S_L. This means that the corresponding transformation maximizes the class variation between channels. Base on the above calculations, the whitening transformation is as follows:

$$W = V^T P$$

In reality we choose the channels corresponding with largest and smallest eigenvalues and ignore the rest of them. Four of these channels were used and the rest was ignored. These channels contain the most useful information for our work. The figure bellow shows the process.



Through testing and cross-validating different features we arrived at the "log(power)" of these four channels as our main features.



For MLP classifier we used 3 hidden layers with "tanh" activation function. The number of neurons in each layer was as followed: [5, 3, 2]

In the case of RBF classifier, we only used ridge regularization. Effectively turning it into an SVM for implementation.[more info] We used the scikit-learn default parameters for the rest. (C=0.1)

Results

Data was split into training and test sets. [85:15]

MLP:

MLP 5-fold cross-validation scores:

[0.87 0.83 0.80 0.75 0.79], avg = 0.81

MLP Test data scores:

0.88

MLP confusion matrix for test data:

[[17, 4], [2, 25]]

Output for the unlabeled data:

[1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1]

RBF:

RBF 5-fold cross-validation scores:

[0.87 0.83 0.81 0.73 0.79], avg = 0.81

RBF Test data scores:

0.88

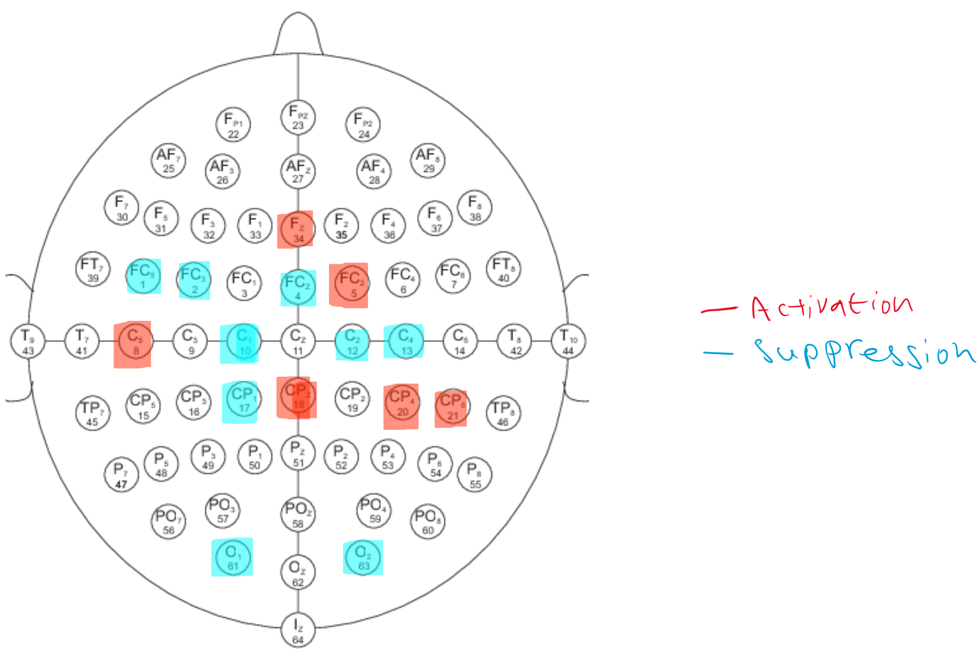
Output for the unlabeled data:

[1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1]

As it can be seen, both RBF and MLP classifier showed relatively good results. Although, RBF was relatively less time consuming to train.

Important Features:

- Log(power) of whitened channels
- The following activation and suppression pattern in beta band was found important



First channel activation and suppression in order

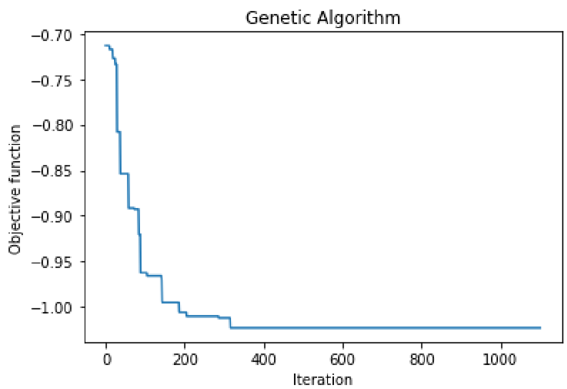
Phase II

In this phase we extract more features from the 4 whitened beta channels and 4 alpha channels and use genetic algorithms to select features. We considered log-power, skew, kurtosis, maximum amplitude, and derivative log-powers. We tested other features too but they had a low separation.

Features = (zt_high_power[1^{st}, 2^{nd}, 3^{rd}, 4^{th} channel], zt_mid_power, zt_high_skew, zt_mid_skew, zt_high_kurt, zt_mid_kurt, zt_high_max, zt_mid_max, zt_high_dot_power, zt_mid_dot_power)

We code our genes as Boolean vectors of length 40 and use a normalized fisher linear discernment matrix to determine fitness(Ratio of traces). The cross-over operation is uniform and the mutation operator is standard. First population is generated at random with a size of 100. We iterate for 1100 generations and keep a 1% elite ratio.

```
algorithm_param = {'max_num_iteration': 1100,\
                   'population_size':100,\
                   'mutation_probability':0.1,\
                   'elit_ratio': 0.01,\
                   'crossover_probability': 0.5,\
                   'parents_portion': 0.3,\
                   'crossover_type':'uniform',\
                   'max_iteration_without_improv':None}
```



The negative of fitness over time

The following gene is the best answer for feature selection. All features corresponding to one are selected for MLP(or RBF) algorithm

[1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

We'll be using PCA to normalize the data before processing them.

MLP architecture will be: (15, 5, 2), with "tanh" activation function

RBF is constructed as before only tuning one parameter.(C=0.5)

Results

Data was split into training and test sets. [85:15]

MLP:

MLP 5-fold cross-validation scores:

[0.93 0.93 0.89 0.77 0.87], avg = 0.88

MLP Test data scores:

0.88

Output for the unlabeled data:

[1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1]

RBF:

RBF 5-fold cross-validation scores:

[0.91 0.87 0.91 0.87 0.91], avg=0.89

RBF Test data scores:

0.92

Output for the unlabeled data:

[0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1]

As it can be seen, both RBF and MLP classifier showed satisfying results. Although, RBF slightly outperformed MLP on the test data and had fewer hyper parameters.

Reference

[1] B. Blankertz, R. Tomioka, et. al: Optimizing Spatial Filters for Robust EEG Single-Trial Analysis (2008)