Implementation of a brain computer interface to classify between two motor activities.

Matjaž Zupančič Muc
Human-Computer Interaction 22/23, FRI, UL
mm1706@student.uni-lj.si

***Abstract*—In this work we implement a brain computer interface (BCI), capable of classifying between two imaginary motor activities, namely opening and closing of the left and right fist. We extracted two types of features from raw electroencephalogram (EEG) signals using common spatial patterns (CSP) and the autoregressive method (AR). We trained two classifiers Linear discriminant analysis (LDA) and Quadratic Discriminant Analysis (QDA). We evaluated models performance on a single subject from the EEG MMIDB database. We were able to overfit the dataset and achieve AUC of 100%, the best AUC using 10 fold cross validation with 30 repetitions was 66.68%.**

I. Introduction

Brain computer interfaces (BCIs) are systems that enable the direct communication between the brain and external devices, allowing individuals to control devices or communicate with the outside world using their brain activity alone. BCIs have the potential to improve the lives of individuals with severe motor impairments, allowing them to perform tasks that would otherwise be impossible. However, developing a BCI that is reliable and easy to use remains a challenge. One important aspect of BCI development is the ability to accurately classify different motor activities. In this work, we present the implementation of a BCI that is capable of distinguishing between two different motor activities. Specifically, we describe the design and evaluation of the BCI system, as well as the results of experiments conducted to test its performance.

II. Related Work

There have been numerous studies on the development and evaluation of BCI systems for a variety of applications. For example, [1] presents a noninvasive BCI that allows users to control a two-dimensional movement signal using their brain activity. [2] describes a BCI system for controlling robotic devices using EEG signals. [3] presents a BCI that allows users to communicate using brain activity alone.

III. Methodology

The input to our system is a set of raw EEG signals of imaginary motor activities, in particular we used subject S001 (records 4, 8, 12) from the EEGMMI Database ([4]). Signals are sampled with a frequency of $Fs = 160$Hz. Each signal in the database is composed of 64 channels. The signal matrix X is therefore of shape $64 \times M$, where M is the number of data points in the signal. Activity intervals are labeled with either T0 (denotes no activity), T1 (denotes imaginary opening and closing of the left wrist) or T2 (denotes imaginary opening and closing of the right wrist). Each interval is roughly 4 seconds long. The interval length is denoted with $m = 4Fs$. In order to extract features from X we used the common spatial patterns (CSP) method. CSP involves projecting the multichannel EEG data onto a set of spatial filters, which are determined such that they maximize the variance of one class while minimizing the variance of the other class. More concretely we sample from X two intervals X_1 and X_2 of size $64 \times m$. Intervals X_1 and X_2 denote the first interval of activity T1 and the first interval of activity T2 respectively. X_1 and X_2 are used to estimate the projection matrix W (of shape 64×64). Intervals that were used to estimate W are removed from the dataset. To compute feature vectors we iterate over intervals X_i and project them into the component space as follows: $S_i = W \times X_i$. The resulting matrix S_i is of shape $(64 \times m)$ and is too large to be used as a feature vector, therefore we keep only the most informative rows (i.e the rows where the variance is maximized), this are the first $K/2$ rows and the last $K/2$ rows. Lets denote the informative matrix (i.e matrix containing only the most informative rows) as $\tilde{S}_i$. Using a band-pass filter we remove noise from $\tilde{S}_i$. We choose a filter which cuts off frequencies outside the $(3-8)$Hz band. Finally we compute the log of Var for each row in $\tilde{S}_i$ as follows: $log(\frac{1}{m}\sum_{j=0}^{m} S_{ij}^2)$. This results in a feature vector f_i of shape $K \times 1$. Figure 1 shows a scatter plot of feature vectors when $K = 2$. Additionally we also compute features using the autoregressive (AR) model. Using only the first and the last row of the filtered $\tilde{S}_i$ we compute model parameters a_i using the Burgs method. Model parameters a_i contain all of the information about the power spectrum of signal $\tilde{S}_i$, therefore they can be used a features. Finally we append a_i to the feature vector f_i. At last we train LDA and QDA using the extracted features.

IV. Experiments

Using subject S001 from the EEGMMI Database ([4]) we compute two sets of feature vectors, the first set contains only CSP features, while the other set contains CSP and AR features. We vary the dimensionality K of feature vectors f_i. For CSP features we tried the following values for K: (2, 4, 8, 16, 22). For CSP and AR features we use the same values for K, but note that the AR features a_i are of length 22, therefore

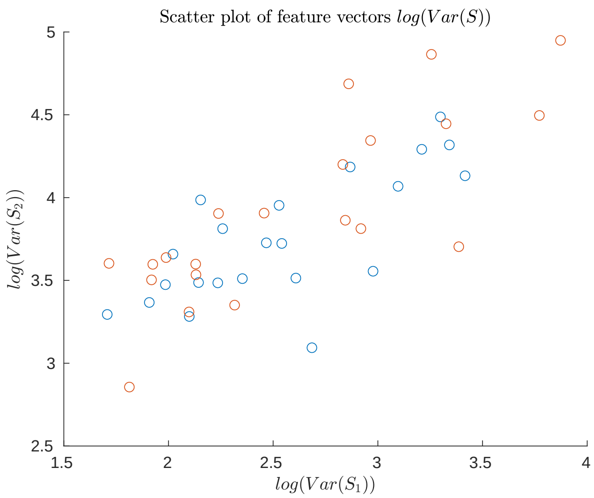

Fig. 1. Scatter plot of CSP features. The blue circles are features which correspond to activity T1, The red circles are features which correspond to activity T2

combined feature vectors have the following number of rows (24, 26, 30, 28, 44).

V. Results and Discussion

A. Results

TABLE I
TRAIN PERFORMANCE USING CSP FEATURES

METHOD	K	Se	Sp	CA	AUC
LDA	2	63.64	57.14	60.47	61.90
QDA	2	77.27	42.86	60.47	63.85
LDA	4	63.64	57.14	60.47	67.97
QDA	4	77.27	66.67	72.09	82.47
LDA	8	81.82	52.38	67.44	77.06
QDA	8	90.91	76.19	83.72	94.59
LDA	16	72.73	80.95	76.74	93.07
QDA	16	100.00	95.24	97.67	100.00
LDA	22	90.91	95.24	93.02	98.27
QDA	22	100.00	100.00	100.00	100.00

TABLE II
TRAIN PERFORMANCE USING CSP AND AR FEATURES

METHOD	K	Se	Sp	CA	AUC
LDA	2	81.82	80.95	81.40	87.45
QDA	2	100.00	100.00	100.00	100.00
LDA	4	100.00	85.71	93.02	99.35
QDA	4	100.00	100.00	100.00	100.00
LDA	8	100.00	85.71	93.02	99.78
QDA	8	100.00	100.00	100.00	100.00
LDA	16	100.00	100.00	100.00	100.00
QDA	16	100.00	100.00	100.00	100.00
LDA	22	100.00	100.00	100.00	100.00
QDA	22	100.00	100.00	100.00	100.00

B. Discussion

Table I shows the "all record Test" performance (i.e the training test performance) using CSP features. We see that as the number of features K increases the over all performance goes up. We see that QDA achivies a higher AUC when comapred to LDA for each K. A perfect performance $Se = 100\%$, $Sp = 100\%$, $CA = 100\%$, $AUC = 100\%$ is achieved for $K = 22$ using QDA. Table II shows the performance using both CSP and AR features. We can see that a perfect performance is achieved using only 2 CSP features in combination with 22 AR features. Generally the performance is overall better when using both CSP and AR features. Table III shows the 10 fold cross validation (using 30 iterations) performance, using CSP features. We can see that the best AUC of 63.16% is achieved using $K = 22$ and LDA. We see that LDA tends to out perform QDA, especially for small values of K, this is likely because QDA is over-fitting the training data. Table IV shows the 10 fold cross validation (using 30 iterations) performance, using CSP and AR features. We can see that the best AUC of 66.68% is achieved using $K = 8$ in combination with 22 AR features and QDA.

TABLE III
30 ITERATIONS OF 10 FOLD CROSS VALIDATION USING CSP FEATURES

METHOD	K	Se	Sp	CA	AUC
LDA	2	59.09	47.62	53.49	51.93
QDA	2	63.64	33.33	48.84	45.69
LDA	4	54.55	52.38	53.49	53.85
QDA	4	54.55	57.14	55.81	60.43
LDA	8	59.09	42.86	51.16	51.20
QDA	8	40.91	42.86	41.86	39.62
LDA	16	54.55	57.14	55.81	59.42
QDA	16	68.18	47.62	58.14	62.19
LDA	22	59.09	57.14	58.14	63.16
QDA	22	45.45	57.14	51.16	52.15

TABLE IV
30 ITERATIONS OF 10 FOLD CROSS VALIDATION USING CSP AND AR FEATURES

METHOD	K	Se	Sp	CA	AUC
LDA	2	36.36	38.10	37.21	36.29
QDA	2	36.36	66.67	51.16	52.25
LDA	4	63.64	59.09	61.36	61.06
QDA	4	40.91	61.90	51.16	55.95
LDA	8	63.64	52.38	58.14	59.63
QDA	8	52.17	66.67	59.09	66.68
LDA	16	50.00	47.62	48.84	47.92
QDA	16	68.18	57.14	62.79	64.50
LDA	22	50.00	47.62	48.84	44.94
QDA	22	56.52	47.62	52.27	51.50

VI. Conclusion

We implemented the CSP and the AR feature extraction method. We evaluated the overall performance and the cross validation performance when using only CSP features and when using CSP with AR. We showed that using both CSP and AR features yields better cross validation AUC compared to only using CSP. Further work could extend the evaluation among multiple patients. We could also add regularization to the QDA classifier, to prevent overfitting.

References

[1] X. Zhang, B. Hong, Z. Jin, X. Gao, and Y. Zhou, "Control of a two-dimensional movement signal by a noninvasive brain–computer interface in humans," *PLoS ONE*, vol. 10, no. 6, p. e0128484, 2015.

[2] Y. Li, X. Gao, Z. Jin, B. Hong, and Y. Zhou, "A brain-computer interface for controlling robotic devices using eeg signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 2, pp. 246–257, 2017.

[3] J. R. Wolpaw, D. J. McFarland, T. M. Vaughan, and E. W. Wolpaw, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[4] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, and H. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.