Passive BCI Hackathon NEC-2021 Submission of team-iBCI

Swapnil Singh, Vartika Gupta, Tharun Kumar Reddy, Vipul Arora [*†]

July 2021

1 Abstract

A passive BCI [ZK11] extracts its outputs from arbitrary brain signal activity occurring in the absence of the objective of voluntary control, for enhancing a human-computer interaction with definite details on the true user state. Over the last decade, passive brain–computer interface (BCI) schemes and bio-signal acquisition technologies have experienced a significant growth that has allowed the real-time analysis of bio-signals, with the objective to quantify pertinent insights, such as cognitive states of the users. However, surpassing reliable proof-of-concepts, employing passive BCIs in day-to-day life requires us to deal with numerous challenges. One such challenge is the within-session variability influencing brain signals such as ElectroEncephaloGraphy (EEG).

However, a significant portion of passive BCI studies were conducted on a single day (also known as session), rendering it unclear if the designed BCI would still function on multiple days/sessions without re-calibration. Since, the EEG signals of every subject and the corresponding label vary with each subject, inter-subject classification using sessions of all the subjects in culmination, with a generalized ensemble of models, has been avoided. In this research, we focus on mental workload classification for a given subject (intra-subject estimation) using the EEG data from another session (inter-session adaptation).

Transfer Learning [WXL20] and Riemannian Geometry [BBCJ13] based methods are state of the art approaches in use, for EEG classification. In this work, we employ Riemannian Geometry based methods.

[*] Swapnil Singh, Vartika Gupta, and Prof. Vipul Arora are with the department of Electrical Engineering at IIT Kanpur (e-mails: swapsing@iitk.ac.in, vartikag@iitk.ac.in and vipular@iitk.ac.in).

[†] Dr. Tharun Kumar Reddy is with Department of Electronics and Communication Engineering at IIT Roorkee (e-mail: tharun.reddy@ece.iitr.ac.in).

2 Feature Extraction

Firstly, a 5^{th} order Butterworth Filter was used to filter raw data within the alpha (8 - 13 Hz) and theta (4 - 7 Hz) frequency bands.

We followed two schemes for electrode selection to decrease the time and space complexity of models without a major loss in data variability. First method involved slicing data for 9 selected electrodes located in the frontal and parietal regions of the brain [GWR+14]. In the second method, we employed Riemannian distance based electrode selection [BB11] which is a subject specific channel selection method. In this approach, pairwise Riemannian distance for every k^{th} class ($k = 0, 1, 2$) is calculated between covariance matrices. The Riemannian distance between two covariance matrices C_1 and C_2 is given by:

$$\delta_r(C_1, C_2) = \|Log(C_1^{-1}C_2\|_F \tag{1}$$

where Log(.) is the log-matrix operator and $\|.\|_F$ is the Frobenius norm of a matrix. A subset of channels are selected and each channel is pruned sequentially to maximize the average Riemannian distance calculated between all pairwise class-conditional matrices as shown in the following criterion (**??**). Here δ_R denotes the pair-wise Riemannian Distance and K (=3) is number of classes

$$\sum_{k=0}^{K-1} \sum_{j>k}^{K} (\delta_R(C_k, C_j)) \tag{2}$$

With validation accuracy for the fine-tuned models as the selection metric, subject-specific channel number was evaluated which addressed data diversity among different subjects. The number of channels for each subject was reduced from 61 to a range of 18-32 channels.

Further, Riemannian Geometry methods were used on the covariance matrices for feature extraction. The covariance matrix C, corresponding to each trial $X \in R^{n_c \times n_s}$, where n_c is the number of channels and n_s is the number of temporal samples, is calculated using the Ledoit-Wolf covariance estimator [LW04] as below:

$$C = (1 - \alpha)A + \alpha \frac{trace(A)}{n_c} I \tag{3}$$

where $\alpha \in [0, 1]$ is the shrinkage factor estimated using Ledoit and Wolf function, I is $n_c \times n_c$ Identity Matrix and A is the simple covariance matrix calculated as:

$$A = \frac{XX^T}{trace(XX^T)} \tag{4}$$

The space of covariance matrices is a differentiable Riemannian manifold M. The derivatives at a matrix C on the manifold lies in a vector space T_P, which is the tangent space at that point. Tangent Space is efficient for classification algorithms like SVM, LDA, Neural Networks which are based on projections into hyperplanes. Each covariance matrix C_i can be mapped into the tangent space located at the geometric mean of the whole set of trials.

$$s_i = upper(C_\theta^{\frac{-1}{2}} Log_\theta(C_i) C_\theta^{\frac{-1}{2}}) \tag{5}$$

where $C_\theta = \theta(C_i, i = 1....K)$ and θ is the Riemannian Mean of covariance matrices calculated as:

$$\theta(C_1, ...C_K) = arg \min_{P \in P(n)} \sum_{i=1}^{I} \delta_r^2(CC_i) \tag{6}$$

Therefore, we calculated tangent space features as the linearization of the manifold of Riemann covariance matrix as described above. The final train and test data is shuffled before passing into the classification models.

3 Classification Results (Model Selection elaborated)

For final classification, Minimum Distance to Mean (MDM), Fisher Geodesic Minimum Distance to Mean (FGMDM) and Deep Neural Network (DNN) classification methods take the covariance features C calculated in (3). Whereas tangent space features calculated in (5), are used as input to the Support Vector Machine (SVM), and Linear Discriminant Analysis (LDA) classification methods (see fig. 1).

The SVM model was implemented with radial basis function kernel and subject-specific regularisation parameter(R). Specific regularisation parameter were obtained by fine-tunning the model for each subject separately to a range of 0.1-10.

The MDM and FgMDM models classify by the shortest Riemannian distance between the test covariance matrix and intra-class covariance matrix means. FgMDM uses geodesic filtering in addition. We employed the Reimannian metric for both the models.

The 5-layer DNN model is a fully connected sequential classification model with Adagrad optimizer, categorical crossentropy loss function and a learning rate of 0.00045, which is iterated for 200 epochs. (512, *relu*), (256, *tanh*), (128, *tanh*), (64, *tanh*) and (3, *softmax*) are the corresponding layers.

The LDA model was used for classification by reducing the dimensionality of input by projecting it to the most discriminative directions using the Singular Value Decomposition solver.

We have also implemented EEGNet model [LSW+18] which takes X as the input. X is the filtered data followed by electrode selection. EEGNet is a 5-layer sequential 2D CNN model with an Adam optimizer and categorical crossentropy as the loss function. The model with a learning and dropout rate of 0.0001 and 0.0025 respectively, and *relu* activation for 4 layers followed by *softmax* activation for the final layer is iterated for 30 epochs. The accuracies achieved can be further improved to achieve comparable results with SVM by applying transfer learning on the model which is a future prospect.

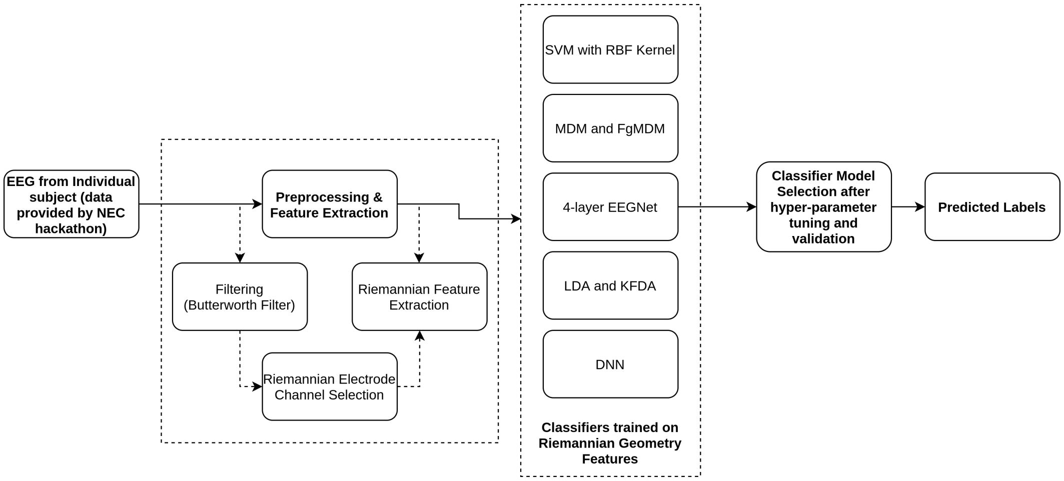
EEG from Individual subject (data provided by NEC hackathon)
Preprocessing & Feature Extraction
Filtering (Butterworth Filter)
Riemannian Feature Extraction
Riemannian Electrode Channel Selection
SVM with RBF Kernel
MDM and FgMDM
4-layer EEGNet
LDA and KFDA
DNN
Classifiers trained on Riemannian Geometry Features
Classifier Model Selection after hyper-parameter tuning and validation
Predicted Labels

Figure 1: Mental Workload prediction Machine learning pipeline

Finally, subject-wise model selection is followed by test labels prediction. The second method shown in table 2 uses modified models and preprocessing which involves channel selection explained in (2) and selective fine- tuning of hyperparameters of SVM, DNN and FgMDM. Since problem is an inter-session adaptation based classification, validation accuracy for each model and subject as shown in table 1 is reported for 2nd session data labels. The SVM classifier with Riemannian channel selection and Riemannian geometry-based covariance matrices performed superior in all subjects except for subject 13 where the DNN classifier performed superior. The average prediction accuracy for all the 15 subjects estimated on the 3rd session came out to be 0.5426, which is a considerable performance in case of inter-session classification methods with limited amount of training data.

In future, we will look at adaptive Riemannian methods for inter-session adaptation and transfer learning methods for the EEGNet model to counter the loss in performance caused by drifts occurring in data across sessions.

Table 1: Inter-session Adaptive Classification Validation Accuracy (*refer to above paragraph below Figure 1)

Subject (along rows) and Methods (along columns)	**Method-1** Filtered Data without channel selection	**Method-1** Unfiltered Data without channel selection	**Modification Parameters (Method-2)*** **(No. of Channels** & **C value for SVM)**	**Method-2** Filtered Data after modification	**Method-2** Unfiltered Data after modification
P01-SVM	0.530201	0.579418	24, 0.1	0.583893	0.60179
P01-LDA	0.304251	0.337808			
P01-FGMDM	0.369128	0.378076			
P01-MDM	0.478747	0.447427			
P02-SVM	0.541387	0.612975	28, 1	0.559284	0.626398
P02-LDA	0.496644	0.536913			
P02-FGMDM	0.46085	0.427293			
P02-MDM	0.46085	0.425056			
P03-SVM	0.630872	0.646532	28, 6	0.637584	0.653244
P03-LDA	0.380313	0.436242			
P03-FGMDM	0.333333	0.344519			
P03-MDM	0.58613	0.545861			
P04-SVM	0.559284	0.630872	32, 4	0.505593	0.657718
P04-LDA	0.431767	0.395973			
P04-FGMDM	0.38255	0.33557			
P04-MDM	0.456376	0.427293			
P05-SVM	0.532438	0.503356	32, 2	0.53915	0.545861
P05-LDA	0.355705	0.456376			
P05-FGMDM	0.375839	0.427293			
P05-MDM	0.47651	0.440716			
P06-SVM	0.702461	0.646532	32, 2	0.727069	0.684564
P06-LDA	0.41387	0.346756			
P06-FGMDM	0.47651	0.510067			
P06-MDM	0.456376	0.454139			
P07-SVM	0.530201	0.563758	28, 0.1	0.604027	0.624161
P07-LDA	0.387025	0.400447			
P07-FGMDM	0.333333	0.46085			
P07-MDM	0.49217	0.503356			

P08-SVM	0.489933	0.619687	28, 4	0.501119	0.646532
P08-LDA	0.38255	0.420582			
P08-FGMDM	0.371365	0.543624			
P08-MDM	0.454139	0.395973			
P09-SVM	0.514541	0.512304	28, 6	0.559284	0.53915
P09-LDA	0.41387	0.487696			
P09-FGMDM	0.315436	0.342282			
P09-MDM	0.512304	0.503356			
P10-SVM	0.626398	0.579418	28, 2	0.63311	0.624161
P10-LDA	0.384787	0.407159			
P10-FGMDM	0.33557	0.348993			
P10-MDM	0.561521	0.46085			
P11-SVM	0.559284	0.552573	28, 2	0.55481	0.588367
P11-LDA	0.454139	0.324385			
P11-FGMDM	0.431767	0.440716			
P11-MDM	0.469799	0.427293			
P12-SVM	0.512304	0.494407	24, 0.1	0.559284	0.521253
P12-LDA	0.420582	0.389262			
P12-FGMDM	0.378076	0.362416			
P12-MDM	0.516779	0.474273			
P12-DNN	0.478901	0.499541			
P12-EEGNet	0.458084	0.437365			
P13-SVM	0.310962	0.416107	24, 0.1	0.404922	0.4443
P13-LDA	0.391499	0.286353			
P13-FGMDM	0.373602	0.310962			
P13-MDM	0.39821	0.418345			
P13-DNN	0.41	0.4443			
P13-EEGNet	0.398	0.417			
P14-SVM	0.519016	0.498881	18, 8	0.49217	0.543624
P14-LDA	0.465324	0.400447			
P14-FGMDM	0.355705	0.387025			
P14-MDM	0.451902	0.378076			
P15-SVM	0.514541	0.545861	22, 2	0.478747	0.565996
P15-LDA	0.364653	0.342282			
P15-FGMDM	0.324385	0.33557			

P15-MDM	0.422819	0.503356			

References

[BB11] Alexandre Barachant and Stephane Bonnet. Channel selection procedure using Riemannian distance for BCI applications. In *2011 5th International IEEE/EMBS Conference on Neural Engineering*, pages 348–351. IEEE, 2011.

[BBCJ13] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172–178, 2013.

[GWR^{+}14] Peter Gerjets, Carina Walter, Wolfgang Rosenstiel, Martin Bogdan, and Thorsten O Zander. Cognitive state monitoring and the design of adaptive instruction in digital environments: lessons learned from cognitive workload assessment using a passive brain-computer interface approach. *Frontiers in neuroscience*, 8:385, 2014.

[LSW^{+}18] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, Jul 2018.

[LW04] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

[WXL20] Dongrui Wu, Yifan Xu, and Bao-Liang Lu. Transfer learning for EEG-based brain-computer interfaces: A review of progress made since 2016. *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[ZK11] Thorsten O Zander and Christian Kothe. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of neural engineering*, 8(2):025005, 2011.