Summary of the paper for NN course: Spatio-Temporal EEG Representation Learning on Riemannian Manifold and Euclidean Space By Guangyi Zhang and Ali Etemad

Bota Duisenbay

June 2022

Contents

1 Summary of the method **1**
1.1 Spatial information 1
1.2 Temporal information 2
1.3 Fusion Strategy 2

2 Implementation details **2**

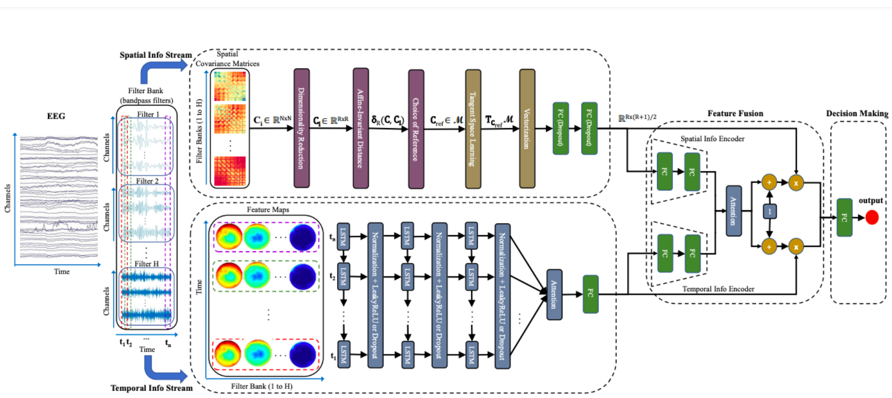
3 Experimental results for BCI IV 2b **4**

1 Summary of the method

The goal of this paper is to learn EEG by studying spatial and temporal information separately and combining the embeddings using effective fusion strategy.

1.1 Spatial information

From multi-channel EEG spatial information is retrieved in the Riemannian manifold and then mapped into Euclidean space. Spatial Covariance Matrices (SCMs) of multiple frequency sub bands are calculated from EEGs. Dimensionality reduction (PCA) is performed to convert computed Symmetric Positive Semi Definite SPSD SCMs into Symmetric Positive Definite SPD SCMs. Further, Riemannian distances between SCMs and mean of SCM are computed Then using tangent space learning method, spatial feature vectors are transferred from Riemannian into Euclidean space, to be further fed into Fully connected layers to learn spatial information.



Dataset	Filter Bank		EEG Trial	Temporal Information Stream		Spatial Information Stream		
Dataset	H	Range	T	L	Features No.	N	Best Rank	Features No.
SEED	5	$1.0 - 50.0Hz$	$8s$	15	10×62	62	48	$5 \times 48 \times (48+1)/2$
SEED-VIG	25	$0.5 - 50.5Hz$	$8s$	15	50×17	17	11	$25 \times 11 \times (11+1)/2$
BCI-IV 2a	25	$0.5 - 50.5Hz$	$4s$	7	50×22	22	18	$25 \times 18 \times (18+1)/2$
BCI-IV 2b	25	$0.5 - 50.5Hz$	$4s$	7	50×3	3	3	$25 \times 3 \times (3+1)/2$

1.2 Temporal information

Temporal information extraction starts from feature extraction: Each EEG from each pre-processed frequency sub-bands is cut into 1 second long Hanning windows with 50% overlap, resulting into L= 7 windows. For each window Density Estimation DE and logarithm Power Spectrum Density features were computed using period-gram method, so that each Hanning window is of size 2xHxN (H number of frequency sub bands and N is number of channels). Extraction temporal features fed into LSTM network: for BCICIV 2a with dropout rates and others with Batch Normalisation and LeakyReLU layer. Finally there are 2 FC layers to learn temporal embedding.

1.3 Fusion Strategy

Each temporal and spatial embedding is fed into FC layers as encoders and united into another FC layer before being fed into a soft attention mechanism. This feature fusion result is further followed by the final FC layer for decision making. Depending on the dataset task , different loss function and activation functions were used. For BCICIV 2b it is binary cross entropy with sigmoid. For each 200 epochs were performed and batch size of 32

2 Implementation details

[1]

[1] Check the code at `https://github.com/botastark/EEG_Riemannian_for_NN_project`

1. conversion.m

   For BCICIV 2a and 2b dataset [1] data were given in gdf format and using MATLAB BioSig toolbox for signal processing, signal and required header information was retrieved and stored in mat format.

2. load.py

   To load signals, true labels and header information from mat files and store it into python readable npy format

3. preprocess.py

   For SEED and SEED VIG sampling rates were down-sampled from 1000Hz to 200Hz, for BCICIV 2a and 2b datasets kept the same (250HZ). For each EEG cut out only relevant portion (for BCICIV 2a/2b from 8 seconds recordings take only between [3.4 - 7.4 seconds] $->$ resulting in total duration T = 4 seconds).

   library/signal_filtering.py: To remove artifacts, 5th order Butter-worth band-pass filters with between 0.5-50.5HZ were applied, as a result sampling at H = 25 sub-bands . It followed by a notch filter at 50 Hz to reduce power line noise. Finally samples were normalized to be in range [-1, 1]. Filtered data stored in separate folder.

4. feature_extraction.py

   For each subject, trial and frequency band, EEG signals were divided into L overlapping Hanning windows and, PSD and DE information were extracted using periodogram method per each channel N and stacked together resulting in 2*H*N features for each window L, frequency band, trail, subject: (trial_n, H, L, 2 * H * N). Extracted features are stored into corresponding folders

5. main_temporal_val.py Extracted features are loaded and divided into k folds. In model/temporal_information.py the features are rearranged into 2d array by [trial * timestep, channel*features], normalized with MinMax scaler to be [-1,1]) and are fed into 3 layer lstm (256 with recurrent activation sigmoid) networks with dropout for BCICIV 2a and Batch Normalization and LeakyRelu for the others. Finally it is fed into 3d attention module composed of 2 dense layers (L size with tanh and softmax activations) before the final FC layer(64 relu). Lose function for SEED and BCICIV 2a is Categorical Cross Entropy, while for BCICIV 2b - Binary Cross entropy and for SEED VIG - Mean Squared Error

6. main_spatial_val.py Depending on number of channels N, it creates N ranks for each dataset. The datasets were divided into K fold and concatenated to for further embedding. Spatial features extraction is implemented in spatial_embedding.py, where first covariance matrices are calculated, then vectorization of the matrices with dimensionality reduction (library/spfiltering.py and library/featuring.py) is performed before mapping from Riemannian to Euclidean space via tangent space learning.

Received Euclidean space features are fed to two FC layers (512 and 64 with relu activation) with 0.5 dropout for BCICIV 2a dataset. Lose function and activations are kept as for temporal embedding. Accuracy and Cohen kappa scores were calculated from difference between predicted and true labels and stored for each rank of the datasets.

For both spatial and temporal embedding following parameters are kept the same and sent as arguments for each function: learning rate=0.001, batch-size = 32 epochs=200 early-stopping=20

7. main.py Fusion is done is 2 steps: first, it encodes spatial and temporal information separately, second, both are fed to soft-attention and decision making FC.

   Stored spatial information is loaded and rank-wise and frequency-band-wise concatenated using (spatial_embedding.py). This is an input to the 3 FC (512, 64 and 1, each with relu activation and 0.5 dropout)

   Extracted features are reshaped to be 2d array [trial*time step, channel*features] and normalized (BCICIV2a as normal distribution with mean and standard deviation, others with MinMax scaler). This is followed by 3 layers of LSTM (256, tanh activation and sigmoid recurrent activation, and [0.2, 0.1, 0.1] dropouts for BCICIV2a) and 2 FC (64 and 1 relu) 3d attention block.

   Results of each encoders concatenated and fed in 3d attention block, that further add 1 element-wise (to shift from scaled in [-1,1] to be positive) and multiplied for each spatial and temporal info and concatenated them back. It is fed to the final FC (128, activation depending on dataset)

3 Experimental results for BCI IV 2b

Avg ± STD	Acc	K
temporal information	0.5068±0.0216	0.0050±0.0350
the paper temporal	0.8073	0.6144
spatial information		
rank 1	0.5354±0.0430	0.0661±0.0827
rank 2	0.5300±0.0406	0.0570±0.0799
rank 3	0.5352±0.0449	0.0668±0.0881
avg	0.5335±0.0428	0.0633±0.0836
the paper spatial	0.8111	0.6217

Table 1: Temporal and spatial results for BCI IV 2b

[1]

[1] The full results are available `https://github.com/botastark/EEG_Riemannian_for_NN_project/tree/main/results/BCI_IV_2b_result`

Subject N	Acc	K
1	0.5187	0.0375
2	0.4964	0
3	0.5031	0.0064
4	0.5031	0
5	0.5188	0.0375
6	0.5059	0
7	0.4984	-0.0004
8	0.4984	0
9	0.5062	0.0125
avg±SD	0.5055±0.008	0.0104± 0.0160
the paper	0.8360±0.1390	0.6720±0.2800

Table 2: Final results for BCI IV 2b