Channel Selection for Motor Imagery Task Classification using Non-linear Separability Measurement

Stuti Chug and Vandana Agarwal[a]

Department of Computer Science and Information Systems, BITS Pilani, Pilani Campus, India

Keywords: Brain Computer Interface, Channel Selection, Non Linearity Measurement, Support Vector Machine.

Abstract: The EEG based motor imagery task classification requires only those channels which contribute to the maximum separability of the training data of different classes. The irrelevant channels are therefore not considered in the formation of feature vectors used in classification. In this paper, we propose a novel algorithm for efficient channel selection (NLMCS). The algorithm computes the proposed metric λ for non-linearity measurement (NLM) and uses this for channel selection. The algorithm is validated on the benchmarked BCI competition IV datasets IIa and IIb. The selected channels are then used for extracting Haar wavelet features and subjected for classification using Support vector Machine. The minimum value of λ corresponds to the optimal channel selection resulting in the best accuracy of motor imagery task classification. The mean Kappa coefficient computed for BCI competition IV IIa dataset using the proposed algorithm is 0.65 and it outperforms some existing approaches.

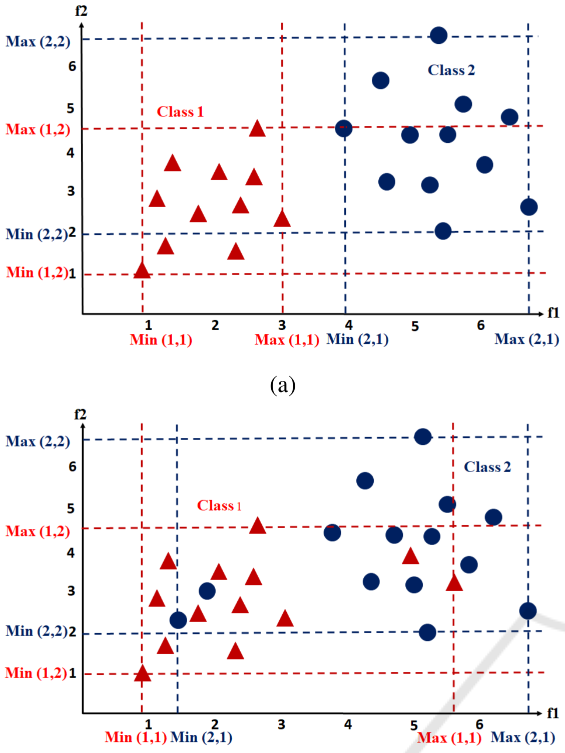
1 INTRODUCTION

Brain computer interface (BCI) systems are used in various application areas such as assistive technology support based on classification of motor imagery of EEG patterns of thoughts, communication and gaming to provide a level of independence to people suffering from neuromuscular disorder. Brain computer interface provides a platform where brain signals are connected with computer. BCI system uses various type of signals such as electroencephalogram (EEG), Electrocorticography (ECog), magnetoencephalography (MEG) and functional magnetic resonance imaging (fMRI). However, EEG signals are widely used in non-invasive BCI system. The basic approach of BCI is used to convert user brain activity pattern into corresponding command. BCI system comprises of different modules such as signal acquisition, preprocessing for artifact and noise removal, channel selection, feature extraction and classification. BCI applications mainly focus on needs of people who are not able to communicate directly to the world e.g. medical applications such as deep brain stimulation for Parkinson disease and cochlear implants for the deaf. BCI introduced an efficient way to convert thoughts into machine commands. Thought patterns have also been examined in applications such as controlling a cursor on the screen by brain, choosing a letter from a virtual keyboard, browsing internet, emotion recognition, motor imagery and gaming (Abdulkader et al., 2015).

The raw EEG signals are captured through a large number of channels, mounted on the scalp of the person, of which some channels are highly active during a particular mental activity such as imagining about the movement of an arm or of a leg. The selection of such active channels providing useful and discriminatory information is a challenging task. These channels once selected, are used to extract the features to represent the thought samples. These samples of EEG signals are represented as points in the highly multidimensional space. These points belonging to different pattern classes may be highly overlapping causing misclassification or may be separable using higher order non linear polynomial functions. Linear separability of these sample points plays important role in classification of thought patterns. If classification data is linearly separable then it is easier to classify as compared to the data which is non-linearly separable. In literature, highly non-linearly separable EEG data is classified using an extremely complex deep neural network that uses a large number of hidden layers (Zhang et al., 2019). This induces an extremely large parametric space to work with, especially for com-

[a] https://orcid.org/0000-0002-8942-5114

Chug, S. and Agarwal, V.
Channel Selection for Motor Imagery Task Classification using Non-linear Separability Measurement.
DOI: 10.5220/0010787300003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 171-178
ISBN: 978-989-758-549-4; ISSN: 2184-4313



(a)

(b)

Figure 1: Sample input space (a) sample data is linearly separable (b) sample data is not separable linearly.

putation of synaptic weights. The optimal parameters thus are computationally intensive and sometimes difficult to be obtained due to the limitations of algorithms such as Gradient Descent and Genetic algorithms, which get trapped in the local optima. In this paper, we attempt to explore the linear separability of the data to be able to work with linear classifiers such as Support Vector Machine to save the time from complex hidden layers and classify thought patterns more efficiently and accurately. Linear separability of data is computed using computational geometry, linear programming, neural network and quadratic programming (Elizondo, 2006; Elizondo et al., 2012).

Our Contribution in This Paper: In this paper, we propose a novel algorithm for Non-Linearity Measurement (NLM) metric to compute the best channels in-order to improve the motor imagery task classification. This algorithm uses the raw EEG data from all channels initially and computes the proposed NLM metric for all classes pairwise for the given combination of channels. The minimum value of NLM metric is then used to select the best channels for classification. The proposed algorithm is validated with the bench marked BCI competition IV datasets IIa and IIb. The algorithm performance has been measured in terms of accuracy and Kappa coefficient, and it outperforms some of the existing work in the same domain.

This paper is organized as follow: Section 2 describes the brief literature survey on channel section, Section 3 presents the basic framework and the proposed algorithm, section 4 discusses the results and section 5 presents the conclusion.

2 RELATED WORK

The aim of using a channel selection algorithm is to enhance the classification accuracy by reducing overfitting issue and also reducing computational complexity while using EEG data. Channel selection is usually considered as part of the feature extraction process.

Channel selection algorithms are based on prior information regarding scalp region of interest of motor imagery task and iteratively optimize the most relevant channels (Shenoy and Vinod, 2014). The authors showed that their algorithm gives 90.77% accuracy on BCI Competition III dataset IVa data using ten channels. A real-time feedback based IterRelCen method using relief algorithm gives 85% accuracy on self-data (Shan et al., 2015). This algorithm is not suitable for the multi-class model and is time-consuming. ReliefF-based channel selection is used for reducing number of channels for convenience in practical usage in DEAP dataset (Zhang et al., 2016). CKSCSP method obtained a minimum set of relevant channels along with the spatial region of brain, with 84.55% accuracy on BCI Competition III dataset (Kirar and Agrawal, 2017). Cohen's d effect size CSP (E-CSP) channel selection algorithm eliminates those channels which do not give any useful information by using z-score (Baig et al., 2020; Das and Suresh, 2015). It shows 85.85% classification accuracy on BCI Competition III dataset IVa by using 9 channels. Sequential Floating Forward Selection (SFFS) technique adds and removes channels according to its Signal Characteristics Advantage Disadvantage P300 (Baig et al., 2020; Qiu et al., 2016). In ICA-BCI system subject specific minimal channel subset was selected to produce high classification accuracy but, in this algorithm, the minimal set of channels is greater than 5-8 (Zhou et al., 2019). A detailed review on channel section techniques can be found in (Baig et al., 2020).

The raw EEG data from selected channels is subjected for feature extraction using various methods such as Fourier transform, Discrete Wavelet transform (DWT) & Haar wavelet (Nicolas-Alonso and Gomez-

Gil, 2012). Various classification algorithms are used for motor imagery classification namely: linear classifier, nonlinear Bayesian classifier, nearest neighbor classifier, neural network and combination of classifiers. Linear classifiers use linear functions for distinguishing the classes such as Linear discriminant analysis (LDA) and support vector machine (SVM) (Gonzalez et al., 2013; Nicolas-Alonso and Gomez-Gil, 2012).

Power spectral density (PSD) technique adopted for feature extraction on various frequency transformations enhances the classification performance using LDA classifier (Alam et al., 2021). In "Distance Preservation to Local Mean" [DPLM] approach for dimensionality reduction the local architecture of the features is preserved by preserving distances to local means (Davoudi et al., 2017). (Gaur et al., 2018) introduced a pre-processing filter approach "Subject Specific Multivariate EMD Filter" [SS-MEMDBF]. The filters based on MEMD reduce the non-stationaries caused by inter and intrasubject differences, thus obtaining enhanced EEG signals. For classification, the author used Riemannian mean computation for all classes. A multi class time-frequency CSP algorithm for motor imagery feature extraction with three different classifiers named Linear Discriminant Analysis (LDA), Naïve Bayes (NVB), and Support Vector Machine (SVM) was proposed by (Zhang and Eskandarian, 2020). Sparse time-frequency segment common spatial pattern (STFSCSP) algorithm is used for features selection in multiple time-frequency segments and the Weighted Naïve Bayesian Classifier (WNBC) was used as classifier (Miao et al., 2017). Bilinear sub-manifold learning (BSML) classification algorithm was proposed to classify the data on a learned intrinsic sub-manifold based on high-dimensional Riemannian manifold (Xie et al., 2016).

3 PROPOSED ALGORITHM

In the present work, we propose to compute the non linear separability of the training samples belonging to different motor imagery classes. Each sample is represented by a d-dimensional point, where d is the total number of features. The j^{th} training sample of p^{th} class is represented by the feature vector T_{pj} given as $T_{pj} = (x^1_{pj}, x^2_{pj}, x^3_{pj}, \ldots x^i_{pj}, \ldots x^d_{pj})$. If the number of intersection points between two classes is less, then the data is highly linear separable. A new metric for NLM for finding the optimal number of channels is proposed. All combinations of the channels are generated and the NLM value for each combination is computed.

Algorithm 1: NLM_pairwise (T^C_a,p,k).

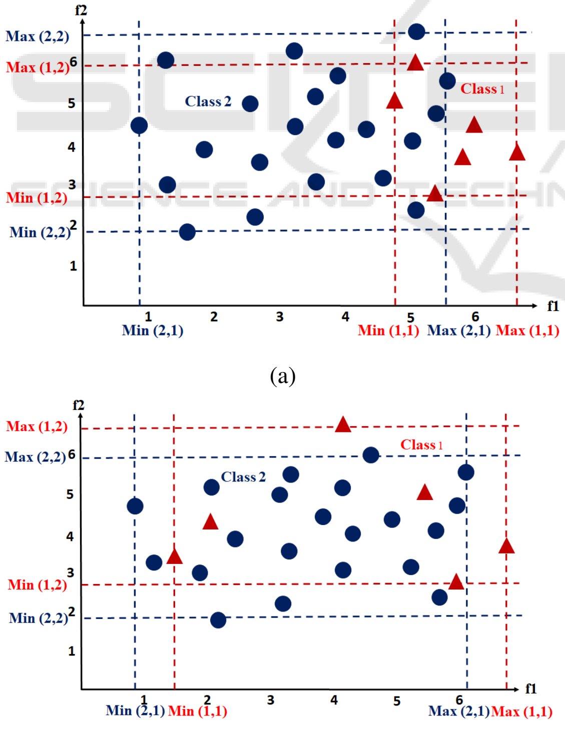
```
Input: $T^C_a$: Feature matrix obtained using
  equation (9), class p, class k
Output: $\lambda^C_{pk}$
D=s*t // D is the total no of features
    from 's' selected channels
$N_{pk}$=0 // Number of overlapping feature
    points between class p & k
for class u = [p,k] do
    for each dimension i = 1 to D do
        $Min(u,i) = Min_j(x^i_{uj})$;
        $Max(u,i) = Max_j(x^i_{uj})$;
for u =[p,k] do
    for v = [p,k] do
        for each training feature vector
          $[T^{C_{r1}}_{auj} T^{C_{r2}}_{auj} T^{C_{r3}}_{auj} \ldots T^{C_{rs}}_{auj}]$ for class u
        count=0 // Initialize count as 0
        if u ≠ v
        for each dimension i = 1 to D do
            if $x^i_{uj} >= Min(v,i) \&\& x^i_{uj} <=$
             $Max(v,i)$ then
                count = count + 1;
        if count == D then
        $N_{pk} = N_{pk} + 1$ // training sample
            adds to overlapping
Compute $\lambda^C_{pk} = \frac{N_{pk}}{N_p + N_k}$ // $N_K$:Number of
    samples in class k
```

Consider a two class sample data represented using 2-dimensional feature vectors [Fig.1]. The class 1 data is represented by red dots and has 11 samples, while class 2 is represented by blue dots which has 12 samples [Fig.1a]. The notion used for sample size of a class p is N_p as is explained after equation (1). In this example, N_1=11 and N_2=12. The term x^i_{pj} represents the i^{th} feature value extracted from the j^{th} sample of the class p where p represents class index ($p = 1,2$), j represents the sample number ($j = 1,2,..m$) and i represents the feature number ($i = 1,2,...d$).

In Fig.1, we have shown samples for two classes $p = 1$ and $p = 2$, where $m = 11$ for $p = 1$, and $m = 12$ for $p = 2$. The axes of coordinate plane are f_1 and f_2. The range of feature values is computed for each class in each dimension along the axes f_1 and f_2 respectively. Each sample data T_{pj} is represented by its feature vector (x^1_{pj},x^2_{pj}) and is shown as a point. The range of class p samples is defined in f_1 dimension by Min(p,1) and Max(p,1), where Min and Max represent the minimum and maximum of all data samples T_{pj} ($j = 1,2$) in f_1 dimension. Similarly, Min(p,2) and Max(p,2) represent the minimum and maximum

Algorithm 2: NLMCS (T).

```
Input: Feature matrix $T$ of training data set
Output: C={$C_{r1}, C_{r2}, C_{r3}, ...C_{rs}$}
All_Channels = [$c_1, c_2, c_3, ...c_q$] // q number
    of channels initially
Min_NLM=10.0; C=All_channels;
    // Initialize
for $a$ = $1$ to $2^q - 1$ do
    ch = $a^{th}$ combination of channel set
      {$C_{r1}, C_{r2}, C_{r3}, ...C_{rs}$} using (7)
    Sum = 0;
    $T_a^c$ = feature matrix obtained using (9) for
      $a^{th}$ channel set
    for $p$= 1: (no. of classes-1) do
        for $k$= p+1: no. of classes do
            $\lambda_{pk}^{ch} = NLM\_pairwise[T_a^c, p, k]$
            $Sum = Sum + \lambda_{pk}^{ch}$
    $\lambda_a = Sum$; // using (2)
    if $\lambda_a <= Min\_NLM$ then
        $Min\_NLM = \lambda_a$; C = ch;
return C // C corresponds to Min_NLM
```



(a)

(b)

Figure 2: The proposed NLM metric is not affected by the class imbalance data. Example class size N_1 = 6, and N_2 = 21. It is affected by the overall spread of overlapping data across other classes. In (a)λ_{12} = 6/(6+21) = 0.222 (b) λ_{12} = 20/(21+6) = 0.74.

values in f_2 direction.

A data sample T_{pj} from class p is investigated if it overlaps with data samples of any other class k (where $k \neq p$) by checking if $x_{pj}^i >=$ Min (k,i) and $x_{pj}^i <=$ Max (k,i) for all $i = 1,2,...d$. If the overlap is in all dimensions, the data sample T_{pj} from class p is considered to be overlapping with class k samples, causing non linear separability. This contributes in the proposed metric (λ_{pk}) of NLM between classes p and k and is defined as follows.

$$\lambda_{pk} = \frac{N_{pk}}{N_p + N_k} \tag{1}$$

Where N_{pk} is the total number of overlapping points between classes p and k, and N_p and N_k are the total number of samples in classes p and k respectively [Algorithm 1]. The proposed metric of non linearity measurement is given below

$$\lambda = \sum_{\substack{p,k=1 \\ p \neq k}}^{n} \lambda_{pk} \tag{2}$$

Where n is the total number of classes in given dataset.

The metric (λ_{pk}) is effectively used for the class balanced data set used in this study. However, the metric does not impose any penalty for the class imbalanced data set [Fig.2]

The feature matrix T of the training data involving all channels prior to channel selection is shown in equation (3).

$$\mathbf{T} = \left[\begin{array}{ccccc}
x_{11}^1 & x_{11}^2 & x_{11}^3 & \cdots & x_{11}^d \\
x_{12}^1 & x_{12}^2 & x_{12}^3 & \cdots & x_{12}^d \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_{1N_1}^1 & x_{1N_1}^2 & x_{1N_1}^3 & \cdots & x_{1N_1}^d \\
x_{21}^1 & x_{21}^2 & x_{21}^3 & \cdots & x_{21}^d \\
x_{22}^1 & x_{22}^2 & x_{22}^3 & \cdots & x_{22}^d \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_{2N_2}^1 & x_{2N_2}^2 & x_{2N_2}^3 & \cdots & x_{2N_2}^d \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_{n1}^1 & x_{n1}^2 & x_{n1}^3 & \cdots & x_{n1}^d \\
x_{n2}^1 & x_{n2}^2 & x_{n2}^3 & \cdots & x_{n2}^d \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_{nN_n}^1 & x_{nN_n}^2 & x_{nN_n}^3 & \cdots & x_{nN_n}^d
\end{array}\right]
\begin{array}{l}
\left.\vphantom{\begin{array}{c}x\\x\\x\\x\end{array}}\right\} \text{class 1: } N_1 \text{ feature vectors} \\
\left.\vphantom{\begin{array}{c}x\\x\\x\\x\end{array}}\right\} \text{class 2: } N_2 \text{ feature vectors} \\
\vphantom{x} \\
\left.\vphantom{\begin{array}{c}x\\x\\x\\x\end{array}}\right\} \text{class n: } N_n \text{ feature vectors}
\end{array} \tag{3}$$

The size of feature matrix T is $r * d$, where r is the number of rows in the feature matrix and is equal to the total number of samples in the training data given as.

$$r = N_1 + N_2 + ... + N_n \tag{4}$$

The number of columns in T is d, which is equal to the total number of channels multiplied by the number of features extracted for a channel [assuming equal number of features extracted from each channel]. Total number of classes is n and N_p is the number of training samples in class p. The raw feature matrix T given in equation (3) involves all channels prior to channel selection. If there are a total of q channels (say $C_1, C_2, C_3 ... C_q$) initially and if t features are extracted from each channel, then the total number of features d is equal to $q * t$. Then the matrix T in equation (3) can further be illustrated as

$$\mathbf{T} = \begin{bmatrix}
\overbrace{x_{11}^{1} \quad \cdots \quad x_{11}^{t}}^{\text{Channel } C_1} & \cdots & \overbrace{x_{11}^{(q-1)t+1} \quad \cdots \quad x_{11}^{qt}}^{\text{Channel } C_q} \\
x_{12}^{1} \quad \cdots \quad x_{12}^{t} & \cdots & x_{12}^{(q-1)t+1} \quad \cdots \quad x_{12}^{qt} \\
\vdots \quad \vdots \quad \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\
x_{1N_1}^{1} \quad \cdots \quad x_{1N_1}^{t} & \cdots & x_{1N_1}^{(q-1)t+1} \quad \cdots \quad x_{1N_1}^{qt} \\
x_{21}^{1} \quad \cdots \quad x_{21}^{t} & \cdots & x_{21}^{(q-1)t+1} \quad \cdots \quad x_{21}^{qt} \\
x_{22}^{1} \quad \cdots \quad x_{22}^{t} & \cdots & x_{22}^{(q-1)t+1} \quad \cdots \quad x_{22}^{qt} \\
\vdots \quad \vdots \quad \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\
x_{2N_2}^{1} \quad \cdots \quad x_{2N_2}^{t} & \cdots & x_{2N_2}^{(q-1)t+1} \quad \cdots \quad x_{2N_2}^{qt} \\
\vdots \quad \vdots \quad \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\
x_{n1}^{1} \quad \cdots \quad x_{n1}^{t} & \cdots & x_{n1}^{(q-1)t+1} \quad \cdots \quad x_{n1}^{qt} \\
x_{n2}^{1} \quad \cdots \quad x_{n2}^{t} & \cdots & x_{n2}^{(q-1)t+1} \quad \cdots \quad x_{n2}^{qt} \\
\vdots \quad \vdots \quad \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\
x_{nN_n}^{1} \quad \cdots \quad x_{nN_n}^{t} & \cdots & x_{nN_n}^{(q-1)t+1} \quad \cdots \quad x_{nN_n}^{qt}
\end{bmatrix} \tag{5}$$

The class labels along with corresponding rows are used in the NLM algorithm for the purpose of computing λ_{pk} for classes p and k. The features matrix in equation (5) above involves all channels which is refined by selecting the channels appropriately. An inclusion of channel consists of retaining the appropriate columns. For example, if the channel C_s is selected, then all t columns are used for the j^{th} sample of class p and is given by

$$\begin{bmatrix} x_{pj}^{(s-1)t+1} & x_{pj}^{(s-1)t+2} & x_{pj}^{(s-1)t+3} & \cdots & x_{pj}^{st} \end{bmatrix}$$

The combination of selected channels of all q channels is represented by the bit string of 0^s and 1^s where 0 represents the absence and 1 represents the presence of a channel in an experimental run. For example a combination 1011001 represents the selection of C_1, C_3, C_4 and C_7 channels. This bit string is also equivalent to a unique decimal number $(64 + 0 + 16 + 8 + 0 + 0 + 1 = 89)$. The position of bit for each channel is fixed. The channel numbering can correspond to either left to right or right to left in the bit string and remains fixed throughout the experiment. In order to form the combinations, we systematically iterate from 1 to $2^q - 1$. This keeps generating equivalent bit string in each iteration which is used to extract the relevant columns from the feature matrix defined in equation (5). A feature matrix T_a^C is obtained by retaining all columns corresponding to the included channels C represented as 1^s in the binary equivalent of the number a. Recalling the above example of a taken as 1011001, the selected channels are written as a set $\{C_1, C_3, C_4, C_7\}$. If $q = 10$ (say) then,

$$\begin{aligned} \{C_1, C_3, C_4, C_7\} \quad \subseteq \quad & \{C_1, C_2, C_3, C_4, C_5, C_6, \\ & C_7, C_8, C_9, C_{10}\} \end{aligned} \tag{6}$$

To generalize, if a total of s channels $\{C_{r1}, C_{r2}, ..C_{rs}\}$ are selected using the binary equivalent of a, then the selected channels can be represented as a subset of all q channels.

$$\{C_{r1}, C_{r2}, ..C_{rs}\} \subseteq \{C_1, C_2 ... C_q\} \tag{7}$$

where $s \leq q$. Let the set of selected channels $\{C_{r1}, C_{r2}, ...C_{rs}\}$, obtained using binary equivalent of a, be represented as C. Then the feature matrix formed by retaining the corresponding t*s number of columns in T from equation (5) can be denoted by T_a^C. Let us define,

$$T_{apj}^{c_{rb}} = \begin{bmatrix} x_{pj}^{(r_b-1)t+1} & x_{pj}^{(r_b-1)t+2} & \cdots & x_{pj}^{(r_b-1)t+t} \end{bmatrix}_{1 \times t} \tag{8}$$

Then,

$$\mathbf{T_a^C} = \begin{bmatrix}
\left.\begin{matrix}
T_{a11}^{C_{r1}} & T_{a11}^{C_{r2}} & \cdots & T_{a11}^{C_{rs}} \\
T_{a12}^{C_{r1}} & T_{a12}^{C_{r2}} & \cdots & T_{a12}^{C_{rs}} \\
\vdots & \vdots & \vdots & \vdots \\
T_{a1N_1}^{C_{r1}} & T_{a1N_1}^{C_{r2}} & \cdots & T_{a1N_1}^{C_{rs}}
\end{matrix}\right\} \text{Class 1} \\
\left.\begin{matrix}
T_{a21}^{C_{r1}} & T_{a21}^{C_{r2}} & \cdots & T_{a21}^{C_{rs}} \\
T_{a22}^{C_{r1}} & T_{a22}^{C_{r2}} & \cdots & T_{a22}^{C_{rs}} \\
\vdots & \vdots & \vdots & \vdots \\
T_{a2N_2}^{C_{r1}} & T_{a2N_2}^{C_{r2}} & \cdots & T_{a2N_2}^{C_{rs}}
\end{matrix}\right\} \text{Class 2} \\
\begin{matrix} \vdots & \vdots & \vdots & \vdots \end{matrix} \\
\left.\begin{matrix}
T_{an1}^{C_{r1}} & T_{an1}^{C_{r2}} & \cdots & T_{an1}^{C_{rs}} \\
T_{an2}^{C_{r1}} & T_{an2}^{C_{r2}} & \cdots & T_{an2}^{C_{rs}} \\
\vdots & \vdots & \vdots & \vdots \\
T_{anN_n}^{C_{r1}} & T_{anN_n}^{C_{r2}} & \cdots & T_{anN_n}^{C_{rs}}
\end{matrix}\right\} \text{Class n}
\end{bmatrix} \tag{9}$$

The size of matrix T_a^C is given as $r * st$, where r is obtained using equation (4), s is the number of channels selected equation (7) and t is the number of features extracted from each channel. Each of $T_{apj}^{C_{rb}}$ in equation (9), for all classes p=1,2,...n and their samples indexed by j=1,2,...,N_p, is then replaced by its corresponding vector as defined in equation (8). The algorithm NLMCS() [Algorithm 2] returns the selected channels to be used for obtaining T_a^c using (9) to train the classifier for motor imagery task classification.

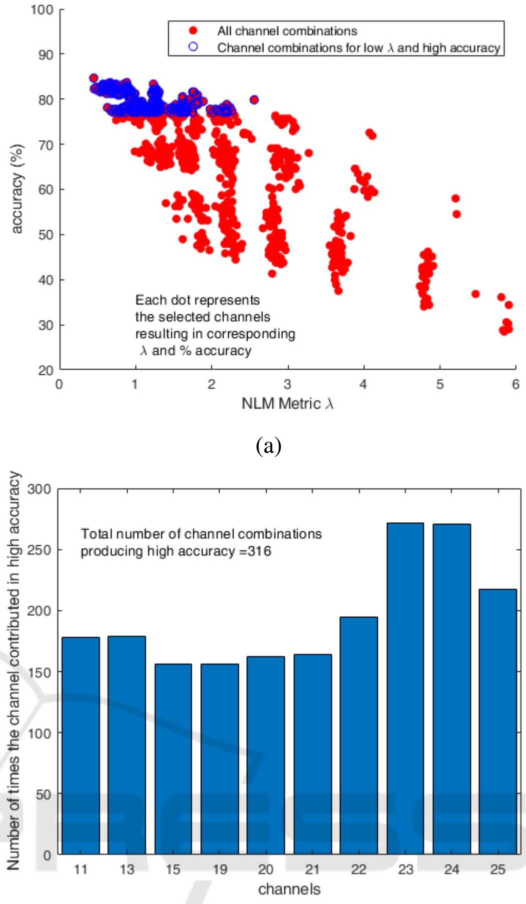
4 EXPERIMENTAL SETUP

4.1 Data Set Description

In this paper, two bench marked datasets namely BCI Competition IV dataset IIa and IIb were used. BCI Competition IV dataset IIa has 4 classes of motor imagery tasks. The EEG signals were collected from nine volunteer participants including four classes namely left hand, right hand, tongue and feet. Two sessions of motor imagery tasks were recorded from each subject, one for training and the other for evaluation. Each session contains 288 trials of data recorded with 25 channels (22 EEG Channels and 3 EOG channels). BCI Competition IV dataset IIb has 2 classes of motor imagery tasks. The EEG signals were collected from nine volunteer participants including two classes namely left hand and right hand movements. Two sessions of motor imagery tasks were recorded from each subject one for training and the other for evaluation. Each session contains 120 trials of data recorded with 3 channels [http://www.bbci.de /competition/iv/] .

4.2 Experimental Evaluation

The performance of the proposed algorithm is evaluated on BBCI Competition IV datasets IIa and IIb as discussed above. The BCI Competition IV IIa dataset has nine subjects as is mentioned above and each subject data is divided into training and testing. We train the model on training dataset and validate the accuracy on testing dataset. Out of the 25 channels, we initially used 10 channels subset (i.e 11,13,15,19,20,21,22,23,24,25) for this experiment to reduce the exponential time complexity. For these 10 channels, 1023 combinations (2^{10}-1) are generated. Thirty features were extracted using Haar wavelet with parameters n=5 and m=4 (where n and m are levels of decomposition) from each of the 10 channels on training dataset. The feature vector size for each training sample class is taken as 600 (60 features* number of channels). The best channels were selected using the proposed NLMCS algorithm and used for classification using Support Vector Machine (SVM). Two performance measures were used to evaluate the performance of the proposed algorithm; cohen's kappa coefficient(k) and accuracy (Acc) computed using the diagonal values of confusion matrix (Table 3). The accuracy (Acc) was computed as the ratio.



(a)

(b)

Figure 3: Channel selection using NLMCS() (a) NLM metric λ versus accuracy scatter graph (b) Number of times channels contributed in high accuracy.

$$Acc = \frac{\sum_{t=1}^{n} n_1^t}{\sum_{t=1}^{n} n_2^t} * 100 \qquad (10)$$

Cohen's kappa coefficient given below is used for evaluating the performance of the proposed algorithm.

$$\kappa = \frac{P_o - P_a}{1 - P_a} \qquad (11)$$

$$P_a = \frac{1}{N^2} \sum_{t=1}^{n} n_1^t * n_2^t \qquad (12)$$

where, P_o and P_a represent observed agreement and chance agreement on test samples respectively. N is the total number of test samples, n is the total number of classes, n_1^t is the total number of samples predicted to be belonging to class t and n_2^t is the total number of samples from actual class t. The proposed algorithm

Table 1: NLM and accuracy of each subject with respective channels selected using proposed algorithm.

	BCI Comp IV dataset IIa												**BCI Comp IV dataset IIb**				
Subject	**Channels**										**NLM**	**Acc(%)**	**Channels**			**NLM**	**Acc(%)**
	11	**13**	**15**	**19**	**20**	**21**	**22**	**23**	**24**	**25**			**1**	**2**	**3**		
Sub 1	0	0	1	1	1	1	1	1	1	1	0.438	61.458	0	1	0	0.870	55
Sub 2	1	1	1	1	0	0	1	1	0	1	0.544	57.916	1	1	1	0.808	66.32
Sub 3	0	1	1	0	0	0	1	1	0	1	0.640	56.944	1	0	1	0.708	60.32
Sub 4	1	1	0	1	0	0	1	1	0	0	0.435	55.55	0	1	1	0.879	54.07
Sub 5	1	1	0	1	0	0	1	1	1	1	0.448	84.680	1	0	1	0.735	67.20
Sub 6	1	1	1	0	0	1	0	1	1	1	0.324	54.166	0	0	1	0.996	65.77
Sub 7	1	1	0	0	1	0	1	1	1	1	0.397	78.819	1	1	0	0.801	61.43
Sub 8	0	0	1	0	1	0	1	1	1	1	0.398	75.694	1	1	1	0.829	52.61
Sub 9	0	0	0	0	0	0	0	1	1	1	0.317	62.152	1	1	1	0.892	56.73

Table 2: Kappa values of the proposed approach and existing approaches for BCI competition IV IIa dataset.

Study	**Approach**	**#C**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Mean**
(Xie et al., 2016)	TSSM+LDA	22	0.77	0.33	0.77	0.51	0.35	0.36	0.71	0.72	0.83	0.59
(Davoudi et al., 2017)	DPLM	22	0.75	0.49	0.76	0.49	0.34	0.36	0.68	0.76	0.76	0.60
(Miao et al., 2017)	DSFTP	22	0.63	0.43	0.74	0.54	0.19	0.26	0.63	0.62	0.69	0.53
(Gaur et al., 2018)	SS-MEMDBF	22	0.86	0.24	0.70	0.68	0.36	0.34	0.66	0.75	0.82	0.60
(Zhang and Eskandarian, 2020)	TFCSP	22	0.62	0.36	0.76	0.40	0.29	0.34	0.59	0.57	0.62	0.51
(Alam et al., 2021)	FE-PSD	22	0.47	0.48	0.92	0.60	0.61	0.50	0.53	0.72	0.68	0.61
Proposed Algorithm	NLMCS	<=10	0.61	0.49	0.61	0.58	0.81	0.53	0.78	0.77	0.67	0.65

Table 3: Confusion matrix for subject 1.

		Predicted class			
		1	**2**	**3**	**4**
Actual class	**1**	**35**	16	13	8
	2	14	**44**	9	5
	3	8	8	**50**	6
	4	7	6	7	**52**

was implemented using MATLAB R2020b and all experiments were performed on Intel(R) Core(TM) i5-4590 CPU Processor(3.30GHz).

4.3 Results and Discussion

This section presents the results of classification accuracy using the proposed NLMCS() algorithm on both BCI Competition IV datasets IIa and IIb. These results indicate that if data is more linearly separable then classification accuracy will enhance. Fig.2(a) displays the effect of NLM on accuracy where each point is a unique combination of channels. The figure depicts the scatter plot of 2^{10} (=1024) points representing unique combinations of 10 different channels. To visually depict the channel selection, in this figure the training data used is of subject 5, while similar plots for other subject are not displayed here to avoid redundant figures. It is observed from the figure that the maximum accuracy (84.680%) is associated with the minimum NLM value (0.448) on one channel combination and the minimum accuracy (29%) is associated with with the maximum NLM value (5.9) on another channel combination. All combinations of channels that attribute to high accuracy (greater than 77% in the figure) and low NLM metric λ (lesser than 6.0) are displayed as blue circles in Fig. 2(a). Of these 316 such different combinations out of total 1024 combinations, a few channels heavily contributed, while other channels were not involved in high accuracy. This observation is presented in Fig. 2(b). It is observed that channels 15 and 19 were least active, while channels 23 and 24 contributed the most for high accuracy. Table 1 summarises the NLM metric (λ) values along with the selected channels to produce maximum accuracy for both the datasets. The accuracy (Acc) and Kappa coefficient (K) where computed using equations (10) and (11).

The proposed algorithm achieved a mean kappa values of 0.65 for all subjects on less than 10 channels. This value was cross-validated on 10 independent runs. The comparative analysis with previous studies of BCI competition IV dataset IIa shows that the proposed algorithm achieves better results compared to other studies in literature as in Table 2. The proposed channel selection algorithm (NLMCS) computation time is 13.107 sec, training computation time through SVM is 0.599 sec and total testing computation time for 288 samples is 2.867 sec. Per sample, the testing time comes out 0.0097 sec.

5 CONCLUSIONS

In this paper, a computational geometry-based algorithm is proposed for selecting the optimal number of channels which is based on non-linear separability measurement. The proposed algorithm is evaluated on both BCI competition IV datasets IIa and IIb. In this work, an effect of low NLM metric (λ) value on high accuracy has been investigated by using the EEG channels that are used to compute such λ. Although, the most effective channels out of the 10 initial channels have been selected, yet the potential of all 25 channels has not been explored. Based on proposed NLMCS() algorithm, we are motivated to apply clustering to compute the λ value for clusters between two classes. In future, the work will be extended for all channels and a subject independent model will be built on features taken from the selected channels design for use in self evolving neural network for EEG classification for improved accuracy.

REFERENCES

Abdulkader, S. N., Atia, A., and Mostafa, M.-S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2):213–230.

Alam, M. N., Ibrahimy, M. I., and Motakabber, S. (2021). Feature extraction of eeg signal by power spectral density for motor imagery based bci. In *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, pages 234–237. IEEE.

Baig, M. Z., Aslam, N., and Shum, H. P. (2020). Filtering techniques for channel selection in motor imagery eeg applications: a survey. *Artificial intelligence review*, 53(2):1207–1232.

Das, A. and Suresh, S. (2015). An effect-size based channel selection algorithm for mental task classification in brain computer interface. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3140–3145. IEEE.

Davoudi, A., Ghidary, S. S., and Sadatnejad, K. (2017). Dimensionality reduction based on distance preservation to local mean for symmetric positive definite matrices and its application in brain–computer interfaces. *Journal of neural engineering*, 14(3):036019.

Elizondo, D. (2006). The linear separability problem: Some testing methods. *IEEE Transactions on neural networks*, 17(2):330–344.

Elizondo, D. A., Birkenhead, R., Gamez, M., Garcia, N., and Alfaro, E. (2012). Linear separability and classification complexity. *Expert Systems with Applications*, 39(9):7796–7807.

Gaur, P., Pachori, R. B., Wang, H., and Prasad, G. (2018). A multi-class eeg-based bci classification using multivariate empirical mode decomposition based filtering and riemannian geometry. *Expert Systems with Applications*, 95:201–211.

Gonzalez, A., Nambu, I., Hokari, H., Iwahashi, M., and Wada, Y. (2013). Towards the classification of single-trial event-related potentials using adapted wavelets and particle swarm optimization. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3089–3094. IEEE.

Kirar, J. S. and Agrawal, R. (2017). Composite kernel support vector machine based performance enhancement of brain computer interface in conjunction with spatial filter. *Biomedical Signal Processing and Control*, 33:151–160.

Miao, M., Zeng, H., Wang, A., Zhao, C., and Liu, F. (2017). Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery eeg: An sparse regression and weighted naïve bayesian classifier-based approach. *Journal of neuroscience methods*, 278:13–24.

Nicolas-Alonso, L. F. and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *sensors*, 12(2):1211–1279.

Qiu, Z., Jin, J., Lam, H.-K., Zhang, Y., Wang, X., and Cichocki, A. (2016). Improved sffs method for channel selection in motor imagery based bci. *Neurocomputing*, 207:519–527.

Shan, H., Xu, H., Zhu, S., and He, B. (2015). A novel channel selection method for optimal classification in different motor imagery bci paradigms. *Biomedical engineering online*, 14(1):1–18.

Shenoy, H. V. and Vinod, A. P. (2014). An iterative optimization technique for robust channel selection in motor imagery based brain computer interface. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1858–1863. IEEE.

Xie, X., Yu, Z. L., Lu, H., Gu, Z., and Li, Y. (2016). Motor imagery classification based on bilinear submanifold learning of symmetric positive-definite matrices. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):504–516.

Zhang, C. and Eskandarian, A. (2020). A computationally efficient multiclass time-frequency common spatial pattern analysis on eeg motor imagery. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 514–518. IEEE.

Zhang, J., Chen, M., Hu, S., Cao, Y., and Kozma, R. (2016). Pnn for eeg-based emotion recognition. In *2016 IEEE international conference on Systems, Man, and Cybernetics (SMC)*, pages 002319–002323. IEEE.

Zhang, R., Zong, Q., Dou, L., and Zhao, X. (2019). A novel hybrid deep learning scheme for four-class motor imagery classification. *Journal of neural engineering*, 16(6):066004.

Zhou, B., Wu, X., Ruan, J., Zhao, L., and Zhang, L. (2019). How many channels are suitable for independent component analysis in motor imagery brain-computer interface. *Biomedical Signal Processing and Control*, 50:103–120.