An Introduction to Common Spatial Pattern for Neural Signal Processing

Ava Lakmazaheri and Nick Steelman

May 2017

Abstract

Common spatial pattern (CSP) is a mathematical procedure used to differentiate between two sets of data. In the context of neuroscience, it is a tool used to maximize the difference between mental states before said data is classified. This paper details the theory and method of computation for CSP and demonstrates its application to signal processing.

1 Introduction

At any instant in the brain, over 100 billion neurons are firing at different frequencies. Extracting and interpreting information from this convoluted network proves to be quite challenging. One of the safest and most cost-effective methods of recording brainwaves, electroencephalography (EEG), attempts to read from noninvasive electrodes that are placed on the scalp. However, this popular method has extremely low spatial resolution and thus cannot be used independently to control complex systems such as brain-computer interfaces. Fortunately, CSP is a utility to further distinguish mental states in EEG data before it is fed into a classifier. Thus, the system will better express the user's intent. In this work, we implemented a CSP filter in MATLAB that demonstrated its ability to increase variance between two classes for multiple sets of data. Sample brainwaves were recorded noninvasively through a Muse brain sensing headband while the user alternatingly focused and relaxed. The filter was deemed successful as the data after CSP appeared visibly more distinguishable than the initial plot.

2 Mathematical Overview

CSP projects temporal information from multiple channels of the EEG into a low-dimensional spatial subspace. The process for accomplishing this is outlined below. The general steps include finding the covariance matrix of each class of data, determining the generalized eigenvalue decomposition of both matrices, and generating the corresponding set of EEG

source distribution vectors (spatial patterns). The final output of this operation is an MxN dimensional matrix where N is the number of channels and M is a selected integer between 1 and N: the number of most important features for distinct classification.

Figures 1 and 2 provide an illustration of the effect of CSP. In this example, the classes are deceivingly easy to distinguish. However, with noisier data, more similar mental states, and fifty additional channels, the data can quickly become impossible for a human to work with alone.

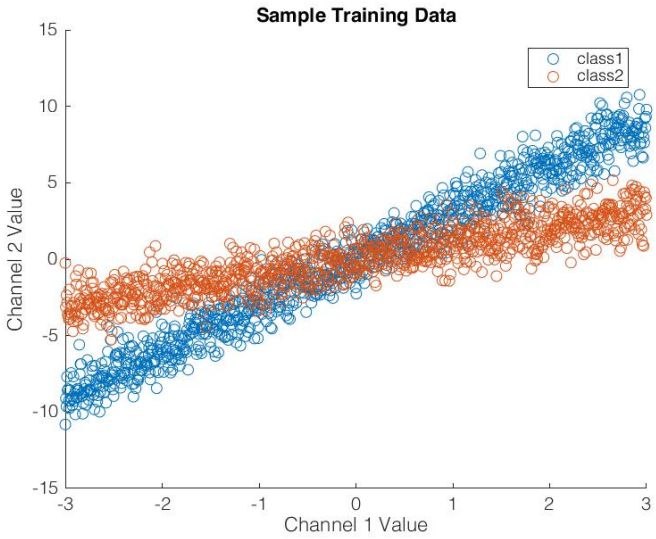

Figure 1: Spatial Plot of Raw Data

CSP takes the above data and compresses it into a non-dependant axis (i.e., an eigenvector) as shown below.

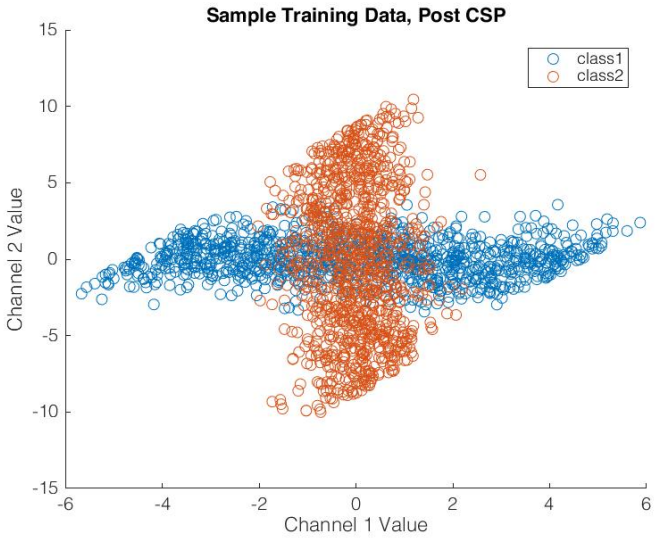

Figure 2: Spatial Plot after CSP Filtering

When this transformed data is squared and averaged, overlap in the classes is even further reduced. At this stage, when fed into a machine learning algorithm, the classification will be significantly more accurate.

3 Implementation

To apply this algorithm, we first recorded two sets of training data using the Muse EEG Headset. The user was instructed to remain alert with their eyes open (labeled state X_1) and then relax with their eyes closed (labeled state X_2), each for thirty seconds. The covariance matrices of each class were computed using the equation:

$$A = \frac{X_1 X_1^T}{trace(X_1 X_1^T)} \qquad B = \frac{X_2 X_2^T}{trace(X_2 X_2^T)} \tag{1}$$

These covariance matrices have a special property: their eigenvectors represent the line of least squares for the data sets, approximating the average vector of each class.

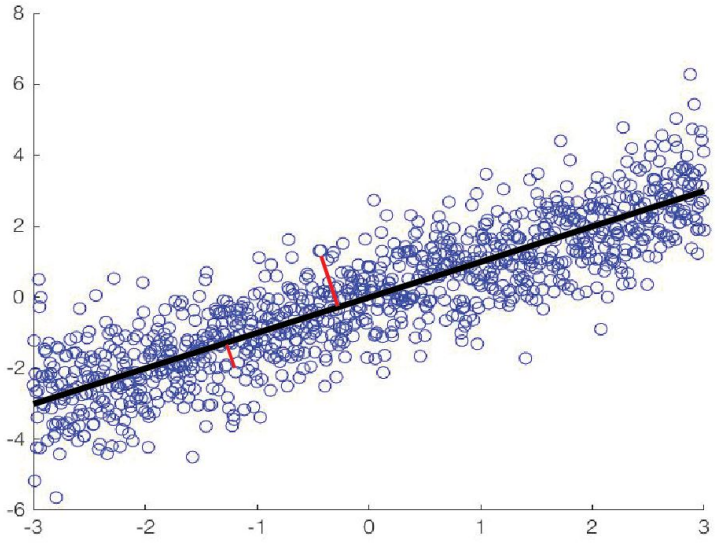

Figure 3: Visual Representation of Covariance Eigenvectors

With the time series data now effectively approximated to a matrix, we want a single transformation that will maximize the difference between the eigenvectors of the covariance matrices. This can be expressed as a change of basis that, when applied to the eigenvectors of both data sets, will result in both matrices having the same eigenvectors but anticorrelated eigenvalues. As such, when the transformation is applied to one data set and maximizes its behavior, it will minimize the behavior of the other class.

To accomplish this, we first summed the two covariance matrices into one matrix R, such that A + B = R. This resultant matrix can be decomposed into its eigenvalues and eigenvectors.

$$R = V_R \lambda_R^T V_R^T \tag{2}$$

Due to the properties of covariance matrices, R has a few special attributes that are essential to understand before moving forward. First, the sum of the eigenvalues for A and B will equal the eigenvalues of R.

$$\lambda_A + \lambda_B = \lambda_R \tag{3}$$

In addition, the eigenvectors of R will be the average of the normal eigenvectors of A and B. Both properties are visually demonstrated in Figure 4.

$$\frac{V_A + V_B}{2} = V_R \tag{4}$$

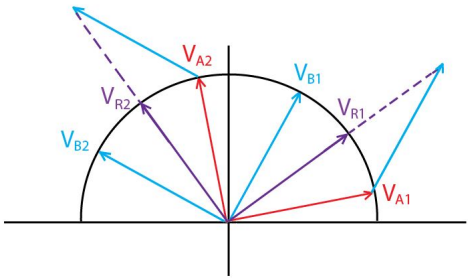

Figure 4: Geometric Properties of Covariance Matrices

With these properties in mind, we define a whitening matrix P as the following:

$$P = \lambda_R^{-1/2} V_R^T \tag{5}$$

The whitening transformation serves to scale all eigenvectors V_R to unit length, decorrelating the covariance axes. Notice that this is roughly the effect in Figure 2, where there is clearly no correlation within the data set.

Next, we define matrices S_A and S_B as the whitened forms of A and B.

$$S_A = PAP^T \qquad S_B = PBP^T \tag{6}$$

$$S_A = \lambda_R^{-1/2} V_R^T A V_R \lambda_R^{-1/2} \qquad S_B = \lambda_R^{-1/2} V_R^T B V_R \lambda_R^{-1/2} \tag{7}$$

Note that S can be simultaneously defined using eigendecomposition.

$$S_A = V_{SA} \lambda_{SA} V_{SA}^T \qquad S_B = V_{SB} \lambda_{SB} V_{SB}^T \tag{8}$$

Defining S_A and S_B in this way forces them to have the same eigenvectors. If you need to convince yourself of that, remember that the eigenvectors of any matrix with a basis change will be the eigenvectors of the matrix multiplied by the basis matrix.

$$V_{SA} = V_R V_A \qquad V_{SB} = V_R V_B \tag{9}$$

Since the eigenvectors of R are the average of the eigenvectors of A and B (see Equation 4 and Figure 4), $V_{SA} = V_{SB}$.

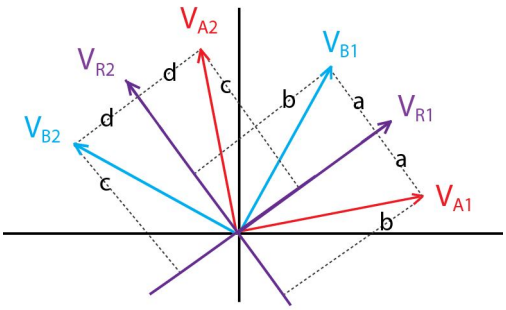

Figure 5: Geometric Transformation of Covariance Matrices

However, a change of basis does not affect eigenvalues. Thus, λ_{SA} and λ_{SB}, eigenvalues that have been scaled by whitening, will sum to the identity matrix.

$$\lambda_{SA} = \lambda_R^{-1/2}\lambda_A\lambda_R^{-1/2} \qquad \lambda_{SB} = \lambda_R^{-1/2}\lambda_B\lambda_R^{-1/2} \tag{10}$$

Referring to previous definitions (Equation 3), this also declares the following.

$$\lambda_{SA} + \lambda_{SB} = \lambda_R^{-1/2}(\lambda_B + \lambda_A)\lambda_R^{-1/2} = \frac{\lambda_B + \lambda_A}{\lambda_R} = I \tag{11}$$

The largest and smallest eigenvalues of either matrix demonstrate the largest difference between the two classes, and the corresponding eigenvector will contain the weights of each channel (i.e., the spatial pattern) that is needed to achieve this distinction.

Finally, we choose the M most prominent patterns and create an M X N weighting matrix. The eigenvectors of A and B are changed to a basis defined by R, are scaled by λ so that the eigenvectors have magnitudes which reflect their weights, and are then projected again onto a basis of the eigenvectors V_{SA}^T such that the largest spread of data sits along the axes.

$$W = V_{SA}^T P \tag{12}$$

Applying W to any new set of data will prepare it for the machine learning classification that follows.

$$Z = WX \tag{13}$$

Thus, we can effectively map a time series of our own recorded brainwaves into a low-dimensional spatial subspace (Figures 6 and 7). Information of this form allows for better signal classification and more accurate brain-computer interfaces.

References: Common Spatial Pattern Method for Channel Selection in Motor Imagery Based Brain-Computer Interface `https://sccn.ucsd.edu/~yijun/pdfs/EMBC05.pdf`

Figure 6: Muse Headset Data showing difference between Open Eyes(Class 1) vs Closed Eyes(Class 2) before CSP is applied

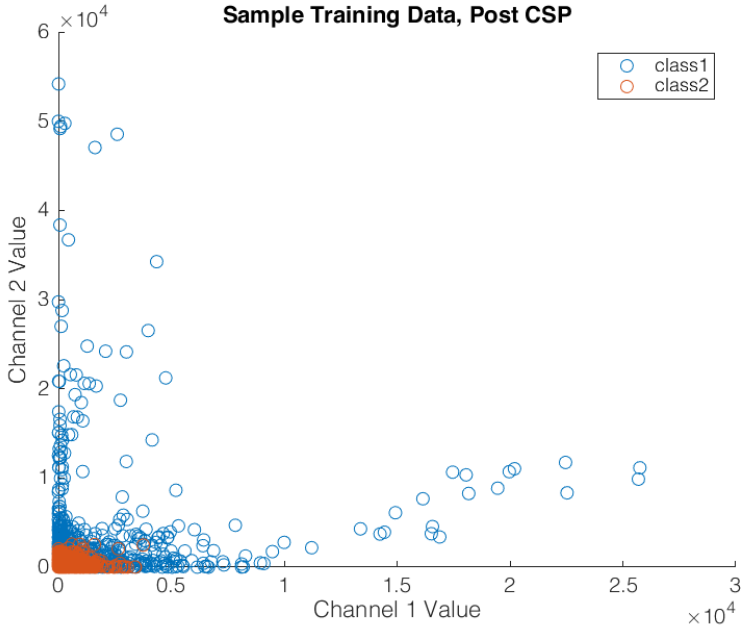

Figure 7: Muse Headset Data showing difference between Open Eyes(Class 1) vs Closed Eyes(Class 2) after CSP is applied