

Combined classification and channel/basis selection with L1-L2 regularization with application to P300 speller system

R. Tomioka^{1,3} and S. Haufe^{2,3}

¹Computer Science, Tokyo Institute of Technology, Tokyo, Japan

²Computer Science, Technische Universität Berlin, Berlin, Germany

³Fraunhofer FIRST.IDA, Berlin, Germany

tomioka@sg.cs.titech.ac.jp,

haufe@cs.tu-berlin.de,

Abstract

We propose a method that combines single-trial classification and channel/basis selection in a single regularized empirical risk minimization problem. We use the linear sum of the Euclidian norms of the columns of the coefficient matrix as the regularizer. This penalty enables us to select rows and columns of the coefficient matrix, which correspond to a subset of the channels or a subset of basis functions, in a systematic manner. Moreover, the parameter learning can be performed in a convex optimization problem with second order cone constraints. The method is demonstrated on P300 speller dataset (dataset II) from the BCI competition III. The method performs reasonably well with small number of electrodes/basis functions.

1 Introduction

Interpretability is one of the key issues in brain-computer interfacing (BCI). A machine learning technique for BCI, though how powerful it might be, can be unacceptable unless it provides insight into the way it functions. Therefore, dimensionality reduction or feature selection techniques, such as independent component analysis or common spatial pattern have become important tools for BCI research (see e.g., [1]). However, these techniques are often developed based on criteria that have little direct connection to the task that we actually need to solve, such as classification or prediction of brain signals. In this paper, we propose a *combined* approach that enables us to reduce the number of channels or basis functions through regularization of the classifier. We use the linear sum of the Euclidian norms of the rows or columns of the coefficient matrix as the regularizer. This regularization enforces the weight matrix to have small number of non-zero *rows* or *columns*. Selecting rows and columns correspond to selecting channels or temporal basis functions. The same idea has also been used in other contexts to achieve joint sparsity of groups of variables (not necessarily rows or columns of a matrix) [2, 3]. The proposed method is applied to P300 speller dataset from the BCI competition III [4] and shows a reasonable performance at small number of channels. Moreover, the inference can be done in a convex optimization problem which can be solved with CVX [5], which is a freely available toolbox for MATLAB.

1.1 P300 speller system

Here we briefly describe the P300 speller system designed by Farwell & Donchin [6]. The subjects are presented a 6×6 table of 36 characters on the screen (see Fig. 1); they are instructed to focus on the characters they wish to spell for some specified period for each character; during that period the rows and columns of the table are intensified in a random order. It is known that the subject's brain shows a characteristic reaction called P300 when the row or column that includes

the character that the subject is focusing is intensified. Thus we can predict the character that the subject is trying to spell by detecting the P300 response. Each intensification lasts 100 ms with an interval of 75 ms afterwards; the intensifications of all 6 rows and 6 columns (in a random order) are repeated 15 times; hence one character takes $175 \text{ ms} \times 12 \times 15 = 31.5 \text{ sec}$. Note that the period of intensification (175 ms) is shorter than the expected reaction of the brain (300 ms). Thus the intervals we analyze are usually overlapped to the next intensification and to each other.

SEND					
A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	-

Figure 1: Table of characters shown on the display in the P300 speller system [6]. The third row is intensified.

2 Method

2.1 Detection model

Let $\tilde{\mathbf{X}} \in \mathbb{R}^{C \times T}$ be a short segment of EEG with C channels and T time-points and let $\mathbf{X} = \mathbf{P}_S \tilde{\mathbf{X}} \mathbf{P}_T$ be the input matrix preprocessed with a fixed spatial and temporal filters, which are defined with regular matrices $\mathbf{P}_S \in \mathbb{C}^{C \times C}$ and $\mathbf{P}_T \in \mathbb{C}^{T \times T}$. Basic goal in many BCI problems is to detect a characteristic spatio-temporal pattern of some activity in \mathbf{X} . Let us write a model for this detection as follows:

$$f_\theta(\mathbf{X}) = \Re(\langle \mathbf{W}, \mathbf{X} \rangle) + b, \quad (1)$$

where $\theta = (\mathbf{W}, b) \in \Theta$ and we call $\mathbf{W} \in \mathbb{C}^{C \times T}$ the coefficient matrix and $b \in \mathbb{R}$ the bias term; $\langle \mathbf{W}, \mathbf{X} \rangle = \sum_{ij} (\mathbf{W})_{ij} (\mathbf{X})_{ij}$ is the inner product between two matrices \mathbf{W} and \mathbf{X} ; $\Re(\cdot)$ denotes the real part of the argument. Note that we use complex coefficients only for the spatio-spectral component selection regularizer (see next section).

2.2 Learning with the L1-L2 regularization

Given n training examples $\{\mathbf{X}_i, y_i\}_{i=1}^n$, where \mathbf{X}_i can be the input matrix or a collection of matrices and $y_i \in \mathcal{Y}$ are the target values we would like to predict, let us introduce the following *regularized empirical risk minimization* [7] problem,

$$\underset{\theta \in \Theta}{\text{minimize}} \quad \sum_{i=1}^n \ell(z_i, \theta) + \lambda \Omega(\theta), \quad (2)$$

where $z_i = (\mathbf{X}_i, y_i) \in \mathcal{Z}$ ($i = 1, \dots, n$). The function $\ell : \mathcal{Z} \times \Theta \rightarrow \mathbb{R}$ is called the *loss function* and it measures how good a parameter configuration θ explains a training example z . The function $\Omega : \Theta \rightarrow \mathbb{R}$ is called the *regularizer* and it measures the complexity of the parameter configuration θ . Our goal here is to minimize the sum of losses that we incur on the whole training examples

and the complexity measured by the regularizer $\Omega(\theta)$. The trade-off between the two terms is controlled by the regularization constant $\lambda (\geq 0) \in \mathbb{R}$.

In this paper we discuss the following three regularizers:

$$\Omega_C(\theta) = \sum_{c=1}^C \|\mathbf{W}(c, :)\|_2, \quad (3)$$

$$\Omega_T(\theta) = \sum_{t=1}^T \|\mathbf{W}(:, t)\|_2. \quad (4)$$

$$\Omega_{CT}(\theta) = \sum_{c=1}^C \sum_{t=1}^T |\mathbf{W}(c, t)|. \quad (5)$$

The first regularizer is called *channel selection regularizer* and it is a linear sum of the Euclidian norms of the row vectors (which correspond to electrodes) of the coefficient matrix \mathbf{W} . The second regularizer is called *basis selection regularizer* and it is a linear sum of the norms of the column vectors (which correspond to time-points) of the coefficient matrix \mathbf{W} . The third regularizer is the sum of absolute values $|\mathbf{W}(c, t)| = \sqrt{\Re(\mathbf{W}(c, t))^2 + \Im(\mathbf{W}(c, t))^2}$ and selects single complex entries of \mathbf{W} and is equivalent to conventional ℓ_1 -norm regularization of the vectorized coefficients (see e.g., [8]) if the coefficient matrix \mathbf{W} is real. It is here called *spatio-spectral component (SSC) selection regularizer*, as we will use it to select pairs of informative electrodes and temporal frequencies.

2.3 P300 decoding model

Let us denote by \mathcal{A} the set of all characters on the screen (see Fig. 1); thus $|\mathcal{A}| = 36$. We denote by $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(12)})$, where $\mathbf{X}^{(l)} \in \mathbb{C}^{C \times T}$, a collection of short segments of EEG recorded after each intensification (1-6 corresponds to columns and 7-12 corresponds to rows). Note that we sort the responses $\mathbf{X}^{(l)}$ ($l = 1, \dots, 12$) according to the indices of rows and columns, which were recorded in the randomized order that the intensifications took place. Additionally let $a \in \mathcal{A}$ be the true character that the subject intend to spell during the intensifications. We formulate the problem of decoding the character $a \in \mathcal{A}$ out of 36 candidates as a direct product of two six-class classification problem as follows:

$$p_\theta(a|\mathbf{X}) = \frac{e^{f_\theta(\mathbf{X}^{\text{col}(a)})}}{\sum_{l=1}^6 e^{f_\theta(\mathbf{X}^{(l)})}} \cdot \frac{e^{f_\theta(\mathbf{X}^{\text{row}(a)+6})}}{\sum_{l=7}^{12} e^{f_\theta(\mathbf{X}^{(l)})}}, \quad (6)$$

where $\text{col}(a) \in \{1, \dots, 6\}$ and $\text{row}(a) \in \{1, \dots, 6\}$ are the indices of the column and the row of the character a on the display (see Fig. 1). Here $f_\theta(\mathbf{X})$ is the linear model in Eq. (1) and outputs a scalar value for each intensification; note that the parameter $\theta = (\mathbf{W}, b)$ is *shared* among all inputs $\mathbf{X}^{(l)}$ ($l = 1, \dots, 12$). In other words, the above model is a direct product of a column classifier and a row classifier in which the scalar value of the model Eq. (1) is interpreted as the degree of P300 response after each intensification. In order to predict a character assuming that we have a detection model $f_\theta(\mathbf{X})$, we maximize the posterior probability $p(a|\mathbf{X})$ given \mathbf{X} with respect to a as follows:

$$\begin{aligned} \hat{a} &= \operatorname{argmax}_{a \in \mathcal{A}} \log p_\theta(a|\mathbf{X}) \\ &= \operatorname{argmax}_{a \in \mathcal{A}} \left(f_\theta(\mathbf{X}^{\text{col}(a)}) + f_\theta(\mathbf{X}^{\text{row}(a)+6}) \right), \end{aligned}$$

which is simply to choose the column and row with maximum response.

2.4 Learning the decoding model

Our training data consists of controlled trials where the subjects are instructed to spell some predefined sequences of characters. Given training examples $\{\mathbf{X}_i, a_i\}_{i=1}^n$, where $\mathbf{X}_i = (\mathbf{X}_i^{(1)}, \dots, \mathbf{X}_i^{(12)})$,

and $a_i \in \mathcal{A}$, our task is to learn the coefficients \mathbf{W} and b . To this end, the most straightforward approach is to use the above decoding model as the conditional likelihood of the training examples. We take the negative logarithm of the likelihood and define the following loss function $\ell(z, \theta)$ as $\ell(z, \theta) := -\log p_\theta(a|\mathbf{X})$, where $z = (\mathbf{X}, a)$. Plugging this loss function into Eq. (2) we obtain the following optimization problem:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n \left\{ -f_{i, \text{col}(a_i)} + \log \left(\sum_{l=1}^6 e^{f_{i,l}} \right) - f_{i, (\text{row}(a_i)+6)} + \log \left(\sum_{l=7}^{12} e^{f_{i,l}} \right) \right\} + \lambda \sum_{c=1}^C u_c, \\ & \text{subject to} && f_{i,l} = \langle \mathbf{W}, \mathbf{X}_i^{(l)} \rangle + b \quad (i = 1, \dots, n, \quad l = 1, \dots, 12), \\ & && u_c \geq \sqrt{\sum_{t=1}^T w_{ct}^2} \quad (c = 1, \dots, C), \end{aligned}$$

where w_{ct} is the (c, t) element of \mathbf{W} , in the case of channel selection regularizer (Eq. (3)); the optimization problem for the basis selection regularizer (Eq. (4)) can be obtained similarly. The above optimization problem is a convex problem with second order cone constraints (see e.g., [9]) and can be solved using CVX toolbox[5] for MATLAB. The above approach that uses the decoding model as the likelihood function for training contrasts sharply with conventional approach that firsts train a binary classifier that detects P300 response and then combines them to predict a character.

2.5 Data acquisition and preprocessing

We use the P300 datasets (dataset II) provided by Jonathan R. Wolpaw, Gerwin Schalk, and Dean Krusienski in the BCI competition III [4]. The dataset includes two subjects namely A and B. The signal is recorded with a multi-channel EEG amplifier with 64 channels. We low-pass filter the signal at 20 Hz, down sample the signal to 60 Hz, and cut out an interval of 600 ms from the onset of each intensification as an *epoch* $\tilde{\mathbf{X}}^{(l)} \in \mathbb{R}^{C \times T}$ where $C = 64$ and $T = 37$ ($l = 1, \dots, 12$). A *trial* $\mathbf{X} \in (\mathbb{C}^{C \times T})^{12}$ consists of 12 preprocessed epoches $\mathbf{X}^{(l)} = \mathbf{P}_S \tilde{\mathbf{X}}^{(l)} \mathbf{P}_T$ ($l = 1, \dots, 12$) and is assigned a single character $a \in \mathcal{A}$. For each character, trials (each consisting of 12 epochs) are repeated 15 times. We average out these repetitions and get 85 training examples ($12 \cdot 85 = 1020$ epochs); although we can also consider each repetition as an individual example in the learning framework (Eq. (2)) as in [10], at the moment we are not able to handle a large training set due to the computational burden. In order to further reduce the training set size we partition 85 trials into 8 sets of 40 trials (480 epochs) regularly overlapped and randomly sampled. We learn \mathbf{W} and b in our discriminative framework with the loss function derived from the decoding model; the outputs of 8 classifiers are simply averaged. Although in [10] it was reported that making an ensemble of SVMs improves performance, such comparison was not possible in our case.

The test data consists of 100 characters; also 12 different intensifications are repeated 15 times (in a random order) in the test set. We report the results of (a) averaging all the 15 repetitions and (b) averaging only the first 5 repetitions in the prediction of each character.

For the channel selection regularizer, we use identity matrices for \mathbf{P}_S and \mathbf{P}_T . For the basis selection regularizer, we use the whitening matrices for \mathbf{P}_S and \mathbf{P}_T as $\mathbf{P}_S = \mathbf{\Sigma}_S^{-1/2}$ and $\mathbf{P}_T = \mathbf{\Sigma}_T^{-1/2}$, respectively. Here $\mathbf{\Sigma}_S$ and $\mathbf{\Sigma}_T$ are the pooled covariance matrices between channels and time-points, respectively, and are defined as $\mathbf{\Sigma}_S = \frac{1}{12n} \sum_{i=1}^n \sum_{l=1}^{12} \text{cov}((\tilde{\mathbf{X}}_i^{(l)})^\top)$, $\mathbf{\Sigma}_T = \frac{1}{12n} \sum_{i=1}^n \sum_{l=1}^{12} \text{cov}(\tilde{\mathbf{X}}_i^{(l)})$, where cov denotes the covariance along the rows (MATLAB `cov` function). Finally, for the SSC selection regularizer, we use the identity matrix for \mathbf{P}_S and for \mathbf{P}_T we use the unitary matrix given by the complex Fourier transform, i.e. $\mathbf{P}_T(t_1, t_2) = \exp(-i2\pi \frac{(t_1-1)(t_2-1)}{T})$ $t_1, t_2 = 1, \dots, T$. As \mathbf{P}_T is complex, we also allow complex coefficients \mathbf{W} , although only real parts enter the loss function. Note that minimization of Eq. (5) is still convex for complex \mathbf{W} and can again be accomplished using second order cone constraints.

3 Results

Figure 2 shows the classification accuracy and the number of active components for channel, basis and SSC selection regularizer from left to right, respectively. Subjects A and B are shown in the left and the right part of each figure, respectively. The top part shows the classification accuracy. Note that a random guess would give $100/36=2.8\%$ accuracy. We can see that the number of non-zero channels, basis, or SSC decreases as the regularization constant λ increase; however the accuracy is almost constant until the regularization shuts off all the coefficients. Figure 3 shows the coefficient matrix topographically mapped on the scalp (nose pointing upwards) for the different regularizers considered. We can see that the coefficients are tightly concentrated around Cz and CPz for the channels selection regularizer (Eq. (3)). On the other hand, since the basis defined by the whitening transformation is fairly localized, the coefficients for the basis selection regularizer (Eq. (4)) are very much concentrated around the time interval from 200 to 300 ms. The frequencies selected by the SSC regularizer are mainly below 5 Hz. The time-courses of the coefficients are therefore very smooth, reflecting the shape of the P300 component. We note that the best accuracy obtained by Rakotomamonjy *et al.* at the competition was 72% and 97% for subject A and 75% and 96% for subject B for 5 repetitions and 15 repetitions, respectively [10].

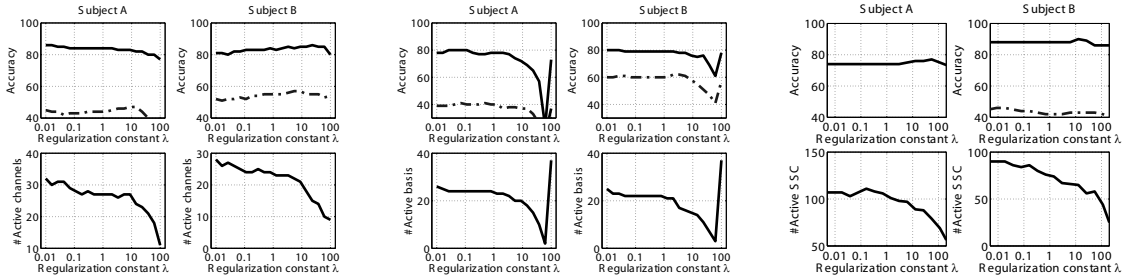


Figure 2: Performance and the number of active channels (left), active basis (center) and active spatio-spectral components (right) with the respective regularizers. In the top part, solid and dash-dotted curves show results of averaging 15 and 5 repetitions, respectively.

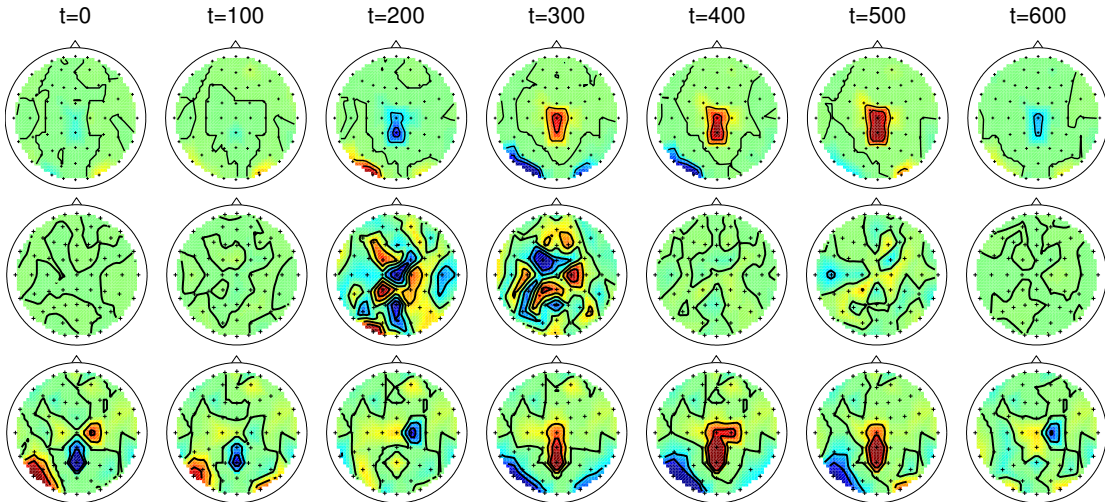


Figure 3: Scalp plots of the coefficients obtained with the along time. Top Row: channel selection regularizer (Subject A; $\lambda = 61.6$). Center Row: basis selection regularizer (Subject B; $\lambda = 8.86$). Bottom row: SSC selection regularizer (Subject B; $\lambda = 12.74$).

4 Conclusion

In this paper, we have proposed a method that performs channel or basis selection jointly with the training of a brain-signal decoding model. We use the linear sum of the Euclidian norms of the rows or columns of the coefficient matrix as the regularizer and perform learning in an empirical risk minimization problem. The regularizer enforces many rows or columns of the coefficient matrix to be simultaneously zero. This row- or column-wise selection contrasts sharply with the element-wise selection that can be obtained with the lasso-regularization [8]. Moreover the training can be done in a convex optimization problem. The proposed method is applied to P300 speller dataset and has shown reasonable performance at small number of electrodes/basis functions. However compared to the best results from the competition the performance was not satisfactory especially when the number of repetition is small. One reason for this might be the fact that we averaged all the repetitions in the training examples and trained the classifier on the averaged data. Thus the classifier underestimates the variability within repetitions. In the future we plan to make the optimization more efficient so that we can handle the original set of training examples without averaging.

Acknowledgments

Ryota Tomioka is supported by Microsoft CORE3 Project. This work was supported in part by grants of the *Bundesministerium für Bildung und Forschung* (BMBF), FKZ 01IBE01A (BCI III) and 01GQ0415 (BCCNB-A4), and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

- [1] Guido Dornhege, José del R. Millán, Thilo Hinterberger, Dennis McFarland, and Klaus-Robert Müller, editors. *Toward Brain-Computer Interfacing*. MIT Press, 2007.
- [2] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.
- [3] Stefan Haufe, Vadim V Nikulin, Andreas Ziehe, Klaus-Robert Müller, and Guido Nolte. Combining sparsity and rotational invariance in EEG/MEG source reconstruction. *NeuroImage*, 2008. In press.
- [4] Benjamin Blankertz, Klaus-Robert Müller, Dean Krusienski, Gerwin Schalk, Jonathan R. Wolpaw, Alois Schlögl, Gert Pfurtscheller, José del R. Millán, Michael Schröder, and Niels Birbaumer. The BCI Competition III: Validating Alternative Approaches to Actual BCI Problems. *IEEE Trans. Neural Sys. Rehab. Eng.*, 14(2):153–159, 2006. See also the webpage: http://ida.first.fhg.de/projects/bci/competition_iii/.
- [5] Michael Grant, Stephen Boyd, and Yinyu Ye. CVX: Matlab Software for Disciplined Convex Programming, July 2007. <http://www.stanford.edu/~boyd/cvx/>, Version 1.1 build 520.
- [6] L.A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysiol.*, 70(6):510–523, 1988.
- [7] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [8] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
- [9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [10] Alain Rakotomamonjy and Vincent Guigue. BCI Competition III : Dataset II - Ensemble of SVMs for BCI P300 speller. *IEEE Trans. Biomed. Eng.*, 55(3):1147–1154, 2008.