# BCI Competition III:

# Dataset II - Ensemble of SVMs for BCI P300 Speller

Alain Rakotomamonjy and Vincent Guigue
LITIS, EA 4108
INSA de Rouen
76801 Saint Etienne du Rouvray, France
Email : alain.rakotomamonjy@insa-rouen.fr

**Abstract**

Brain-Computer Interface P300 speller aims at helping patients unable to activate muscles to spell words by means of their brain signal activities. Associated to this BCI paradigm, there is the problem of classifying electroencephalogram signals related to responses to some visual stimuli. This paper addresses the problem of signal responses variability within a single subject in such Brain-Computer Interface. We propose a method that copes with such variabilities through an ensemble of classifiers approach. Each classifier is composed of a linear Support Vector Machine trained on a small part of the available data and for which a channel selection procedure has been performed. Performances of our algorithm have been evaluated on dataset II of the BCI Competition III and has yielded the best performance of the competition.

## 1 Introduction

Some people who suffer neurological diseases can be highly paralyzed and incapable of any motor functions but still have some cognitive abilities. Their only way to communicate with their environment is by using their brain activities. Brain-Computer interfaces (BCI) research aims at developing systems that help those disabled people communicating through the use of computers and their brain waves.

Research on BCI is a fast growing field and several electroencephalogram (EEG) based techniques have been proposed for realizing BCI. The BCI framework that interests us is based on P300 Event Related Potentials (ERP) which are natural responses of the brain to some specific external stimuli. Such a BCI produces an appropriate stimulus for which the patient is expected to respond. The core principle underlying such a BCI system is then the ability of the system to correctly recognize ERP that are responses to stimuli. In other words, such a BCI system is wrapped around a classification technique. Within the context of P300 based BCI, several classification methods, like support vector machines or linear discriminant analysis, have proven their efficiency [8, 4, 17].

Since few years now, several BCI competitions have been organized in order to promote the development of BCI and the underlying data mining techniques. For instance, a more detailed overview of the BCI competition II and III are described in the papers of Blankertz et al. [2, 3]. These competitions allow the community to benchmark several classification techniques in an unbiased way. Indeed, development and test data are provided by BCI laboratories but the truth about test data are not known by competitors. Such competitions are thus of great interest since they give hints on classification approaches that "work well".

This paper presents the algorithm that has provided the best classification performance on the dataset produced by a P300 speller matrix during the BCI III competition. Naturally, one drawback of such paper is that it provides only some offline analysis of the classification algorithm and online capacity still has to be verified. Furthermore, BCI competition III has only provided datasets from 2 different subjects although from different acquisition sessions. Despite such limitations, we believe that this paper provides an interesting contribution in the area of classifier for BCI especially because the results that we expose have been validated in an unbiased way. Furthermore, in order to enhance the impact of our paper and for a sake of reproducibility, the code of the algorithm has been made available at : `http://asi.insa-rouen.fr/~arakotom/code/bciindex.html`

The paper is structured as follows : section 2 shortly describes the BCI data set provided by the competition. In section 3, we detail the methodology that has been followed which is based on an ensemble of SVMs and a channel selection procedure. Finally, section 4 presents the results we achieved and section 5 concludes the paper with comments and perspectives on the work.

# 2 The data set

## 2.1 Description

For this competition, we have focused only on recorded brain signals produced by the BCI P300 speller problem. The P300 speller is based on the so-called *oddball paradigm* which states that rare expected stimuli produce a positive deflection in the EEG after about 300 ms. This so-called P300 component is present in nearly every human. A P300 speller, based on this paradigm, has been introduced by Farwell and Donchin [5] who developed a protocol whereby a subject is presented a $6 \times 6$ character matrix as illustrated in Figure 1. For the spelling of a single character, each of the 12 rows and columns of the matrix is then intensified according to a random sequence (in the sequel, we denote as a sequence such a set of 12 intensifications). The subject is asked to focus its attention on the character he wants to spell and then a P300 evoked potentials appear in the EEG in response to the intensification of a row or column containing the desired character. In order to make the spelling procedure more reliable, this sequence of intensifications is repeated 15 times for each character to spell.

For this competition, the dataset, which is still available on the competition webpage [1], has been recorded from two different subjects and 5 different spelling sessions. Each session is composed of runs, and for each run, a subject is asked to spell a word. For a given acquisition session, all EEG signals of a 64-channel scalp (see Figure 1) have been continuously collected. Before digitization at a sample rate of 240 Hz, signals have been bandpass-filtered from $0.1 - 60$ Hz [13]. A more detailed description of the dataset can be found in the BCI competition paper [3].

The classification problem we address is the following : given the 64-channel signals collected after the intensification of a row or column, named a post-stimulus signal, we want to predict if such signal contains or not a P300 ERP. This first part of the problem is then a binary classification problem. Afterwards, according to the classification of each post-stimulus signal,

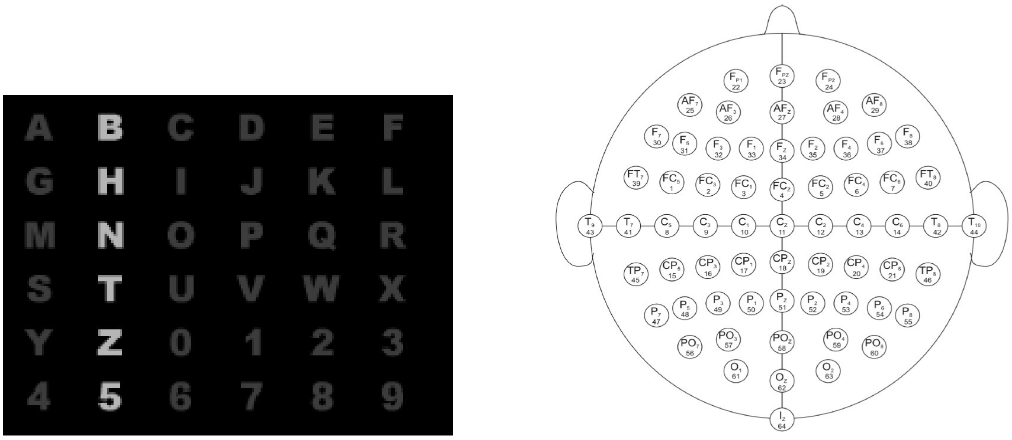
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 0 1 2 3
4 5 6 7 8 9
$F_{P1}$ 22 $F_{PZ}$ 23 $F_{P2}$ 24
$AF_7$ 25 $AF_3$ 26 $AF_Z$ 27 $AF_4$ 28 $AF_8$ 29
$F_7$ 30 $F_5$ 31 $F_3$ 32 $F_1$ 33 $F_Z$ 34 $F_2$ 35 $F_4$ 36 $F_6$ 37 $F_8$ 38
$FT_7$ 39 $FC_5$ 1 $FC_3$ 2 $FC_1$ 3 $FC_Z$ 4 $FC_2$ 5 $FC_4$ 6 $FC_6$ 7 $FT_8$ 40
$T_9$ 43 $T_7$ 41 $C_5$ 8 $C_3$ 9 $C_1$ 10 $C_Z$ 11 $C_2$ 12 $C_4$ 13 $C_6$ 14 $T_8$ 42 $T_{10}$ 44
$TP_7$ 45 $CP_5$ 15 $CP_3$ 16 $CP_1$ 17 $CP_Z$ 18 $CP_2$ 19 $CP_4$ 20 $CP_6$ 21 $TP_8$ 46
$P_7$ 47 $P_5$ 48 $P_3$ 49 $P_1$ 50 $P_Z$ 51 $P_2$ 52 $P_4$ 53 $P_6$ 54 $P_8$ 55
$PO_7$ 56 $PO_3$ 57 $PO_Z$ 58 $PO_4$ 59 $PO_8$ 60
$O_1$ 61 $O_Z$ 62 $O_2$ 63
$I_Z$ 64

Figure 1: Experimental set-up. left) Example of a 6 × 6 user display in P300 Speller. right) Channel location and assignment numbers used for EEG acquisition [3].

our aim is to correctly predict the desired character using the fewest sequences as possible. Hence, this second part of the problem deals with a 36-class classification problem since we want to recognize a symbol from the 6 × 6 matrix as given in Figure 1. Recognition rate of spelled characters is the evaluation criterion of the competition.

For solving this problem, we are provided a training set of signals for which the target characters are known. Note that these characters come from the spelling of word, but they have been scrambled so that spelling time chronology has been lost. For each subject, the training set is made of 85 characters spelling which correspond to $15300 = 12 \times 15 \times 85$ post-stimulus labeled signals (each of them collected over 64 channels).

## 2.2 Data preprocessing and feature extraction

Since we are only interested in part of the EEG signals occurring after each intensification and because we want to build some "high-level" features that can be fed to a classifier, these signals have been preprocessed.

At first, for each channel, we extracted all data samples between 0 to 667 ms posterior to the beginning of an intensification. According to the knowledge that the evoked potentials appear about 300 ms after the stimulus, we postulate that this window is large enough to capture all required time features for an efficient classification. Afterwards, each extracted signal has been filtered with an 8-order band-pass Chebyshev Type I filter which cut-off frequencies are 0.1 and 10 Hz and has been decimated according to the high cut-off frequency. At this point, an extracted signal from a single channel is composed of 14 samples.

Figure 2 gives an example of the sample variations of positive (with P300) and negative (without P300) signals after preprocessing. The left plot presents the mean signals (averaged over all training signals) for the channel $PO_z$ whether P300 ERPs are present or not. The right plot illustrates all the difficulty of the classification problem. For this plot, the three panels present the 7th and 8th variables for the channels $P_z$, $PO_7$ and $PO_z$. Then, for each channel and variables, we compare the left and right boxplots of positive and negatives examples after normalization. Samples 7 and 8 of each channel have been plotted because we expect that they are related to events occurring around $300 - 350$ ms. As we can see, signals with P300 clearly have, on average, larger values of samples 7 and 8. However, sample variabilities are also so large to make the classification of a signal difficult.

After this preprocessing stage, a post-stimulus signal has been transformed into a vector from the concatenation of the 14 samples of all the 64 channels. Thus, for a single subject, the training set is composed of 15300 post-stimulus vectors $x_i$ of dimension $896 = 14 \times 64$ for which labels are $y_i = \{1, -1\}$.

## 3 Methods

In this section, we present the methodology we followed for building our classifier. At first, we describe our multiple classifier strategy for the binary classification problem. The second subsection then deals with how all binary classification results are transformed into a decision for the 36-class problem. The next 2 subsections then present the channel selection and the model selection procedure used for each of the single classifier. Note that in our classification strategy, a classifier for a given subject only uses training data generated by that subject.

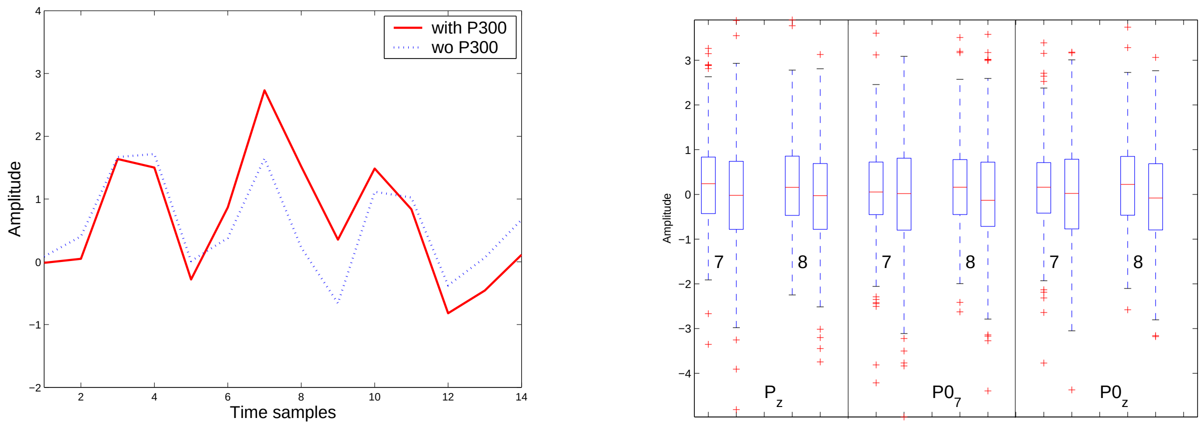

Figure 2: left) The 14-sample signal averages for the channel $PO_z$. Curve for positive signal average (with P300) is depicted in solid line whereas curve for negative signal average (without P300) is shown in dotted line. right) Boxplots of the 7th and 8th variables associated to channels $P_z$, $PO_7$ and $PO_z$. For each channel and variables, the left boxplot denotes the one of positive examples while the right boxplot is related to negatives examples. Note that for building these boxplots, all variables have been normalized to zero mean and unit variance. We can see that on average, positive examples have a higher mean values than negatives ones for the considered variables and channels.

### 3.1 Ensemble of classifiers strategy

If we make the hypothesis that signal variability is essentially related to spelling-unrelated EEG signal components or other brain activities, then a way to reduce such a variability is to perform signal averaging which is a classical method for enhancing signal-to-noise ratio. We will make clear latter how we perform such an averaging.

Another way to reduce the influence of signal variability in this classification problem is to use an ensemble of classifiers approach. Indeed, it is well known that techniques based on classifier outputs averaging help to reduce classifier variability [6].

When using such an ensemble of classifiers system, each single classifier has its own training data. Hence, we have to partition the 15300 examples of the training set in different subsets.

Ideally, it may be interesting to cluster the training signals in several partitions so that each partition has "similar" noisy components. Consequently, we suggest separating the training signals into homogeneous groups and to train a classifier on each of these groups. To this end, the notion of "homogeneous" has to be defined. The simplest approach is to consider that the 180 post-stimulus signals related to the spelling of a single character present similar noisy features since they have been acquired in a relatively short period of time. Indeed, during this short time, we can suppose that brain activities and acquisition conditions have not changed considerably. On another level, signals from the same acquisition run (coming from the spelling of the same word) can also be considered as homogeneous.

For the dataset we have been provided, time chronology of spelled characters has been lost because organizers decided to scramble them [3]. Hence, signals composing the training set can not be naturally clustered together at a higher level than character spelling since all information about runs and sessions have been lost.

Hence, we have decided to use the following naive partitioning. Single character spelling corresponds to 180 post-stimulus signals. Each training partition is then composed of signals associated to 5 characters, leading to 17 different partitions. Since time chronology of signals have been lost, the 5 characters composing each partition are each 5 consecutive characters as provided by organizers. According to this procedure, for each subject $A$ or $B$, $K = 17$ different partitions have been built. They are denoted as $A_1$, $A_2$, $\cdots$, $A_{17}$ and $B_1$, $B_2$, $\cdots$, $B_{17}$. Each partition is then composed of 900 training examples of dimension 896.

In our strategy, we have designed a multiple classifier system for each single subject. Each single classifier of the system is a linear support vector machine [14] trained on one of the 17 partitions. SVM has been used since it is a powerful approach for pattern recognition especially for high-dimensional problems. Each single SVM training also involves a model selection procedure for setting its regularization parameter $C$. We will give details about this part in the sequel. The decision function for an SVM trained on the $k$-th partition $P_k$, with $P_k$ being either $A_k$ or $B_k$, is then :

$$f_k(x) = \sum_{i \in P_k} y_i \alpha_i^{(k)} \langle x, x_i \rangle + b^{(k)} \tag{1}$$

where the $\{\alpha_i^{(k)}\}_i$ and $b^{(k)}$ are parameters obtained after SVM training. Performance of each classifier on a new post-stimulus signal $x$ (also known as single trial classification) can be evaluated by looking at the sign of $f_k(x)$.

### 3.2 Global classification scheme

We now explain how outputs of these classifiers are fused together in order to produce a single predicted character. Thus, we are transforming the results of all binary classifiers into a single result on the 36-class problem.

Our classification method is based on $K = 17$ classifiers for which each classifier has been trained on a subset of the training data. Each classifier assigns a real-valued score $f_k(x_{r|c})$, $k = 1, \cdots, K$ to a post-stimulus vector $x_{r|c}$ associated to a given row or column. After a number of sequences $J$, we consider that the most probable row and column is the one that maximizes the score :

$$S_{r|c} = \frac{1}{J}\frac{1}{K}\sum_{j=1}^{J}\sum_{k=1}^{K} f_k(x_{r|c}^{(j)}) \tag{2}$$

where $x_{r|c}^{(j)}$ is the post-stimulus vector associated to a given row or column during the $j$-th sequence and $S_{r|c}$ is the score of that row or column. Expanding this equation shows that :

$$S_{r|c} = \frac{1}{K}\sum_{k=1}\sum_{i\in P_k} y_i\alpha_i^{(k)}\left\langle \frac{1}{J}\sum_{j=1}^{J} x_{r|c}^{(j)}, x_i\right\rangle + b^{(k)} \tag{3}$$

which clearly exhibits the double averaging performed by our classifier. The first averaging is applied in the data space : as sequences increase, signals from each row or column are averaged. The second averaging is done in the classification score space. This latter procedure leads to a more robust classification scheme since a classifier that assigns a bad score to a test data can be corrected by other classifiers.

Note that the idea of averaging SVM outputs over the sequences as in equation 2 has already been successfully applied by Kaper et al. [8] (but without the ensemble of SVM) and they yielded very interesting performances. However, since they use a non-linear classifier, a Gaussian kernel SVM, their averaging does not boil down to signal averaging as in our approach (see equation 3) .

### 3.3 Channel selection procedure

The dataset provided by the competition is based on a 64-channel scalp. However, in many P300 BCI classification systems, it is frequent that only a limited number of channels is used. For instance, Meinicke et al. use only 10 predefined channels to build their P300 BCI [10] while Serby et al. use only 3 of them [16]. Although the need of reducing the number of channels is clear, we advocate that such channels have to be selected adaptively with respects to the subject and the mental task to be performed, hence the need of automatic channel selection algorithm. Algorithms for channel selection can identify, among these 64 channels, the most efficient ones for revealing presence of P300 ERP. Thus, channel selection can help at reducing the number of electrodes needed for building each decision function while increasing recognition performance by removing spurious channels.

The algorithm we use is based on a recursive channel elimination. In this algorithm, classifier performances are frequently evaluated on a validation set according to the score:

$$C_{cs} = \frac{tp}{tp + fp + fn}$$

where $tp, fp, fn$ are respectively the number of true positive, false positive and false negative classified post-stimulus signal composing the validation set. It is important to note that i) for channel selection, classifier performances are evaluated on a single post-stimulus signal (binary classification) and not on character recognition performance. ii) the score $C_{cs}$ does not take into account the number of true negative examples. This is important for unbalanced datasets since this omission helps the channel selection procedure to focus on channels that give positive scores to positive examples which are fewer than negative examples.

The algorithm we used is given by the following procedure. First, a linear SVM is trained with all the 896 features provided by all channels. The performance of this classifier is evaluated according to $C_{cs}$. Then, each single channel is temporarily removed, (suppose we remove channel $j$,) that is to say all 14 features built from that channel are removed, and the score $C_{cs}^{(-j)}$ (the score when channel $j$ is removed) is re-evaluated. Finally, the channel whose removal has maximized the score $C_{cs}^{(-j)}$ is definitely eliminated. The procedure is then continued until all channels have been eliminated. In order to speed-up this elimination procedure, it is possible

**Algorithm 1** Algorithm for recursive channel elimination.

```
Initialization : RANKED= $\emptyset$; CHANNEL= $[1, \cdots, \text{NumberOfChannels}]$
while CHANNEL is not empty do
  for all channel in CHANNEL do
    Remove temporarily channel $j$ in CHANNEL
    Learn a linear SVM with the remaining channels
    Compute ranking score $C_{cs}^{-(j)}$
  end for
  RANKCHAN= $\arg\max_i C_{cs}^{-(j)}$
  Rank variable : RANKED = [ RANKCHAN RANKED ]
  Remove variable RANKCHAN from the variable list CHANNEL
end while
```

to eliminate that several channels after each step. In our case, we have arbitrarily chosen to remove four channels at a time. This channel elimination procedure is also a way for ranking the channels according to the score $C_{cs}$, the first eliminated channel being the less important one and the last removed being the most important one. However, note that this ranking procedure is suboptimal since a channel that has been eliminated is never questioned again in the elimination procedure. The algorithm for such recursive channel elimination is described in Algorithm 1. More details on similar variable selection procedure can be found in [7, 11].

### 3.4 Model selection

For each of the 17 linear SVM we train, a model selection procedure has been performed. In our case, the model selection involves the choice of the classical SVM hyperparameter $C$ and the optimal number of channels to use. Hence for such model selection procedure, since each classifier is trained on one of the 17 partitions, we have used as a validation set a subset of the 16 remaining dataset partitions (technical details on how validation sets have been built are postponed to the appendix). Prior to training, all features from a given training set have been normalized to zero mean and unit variance. Validation set has also been transformed according to the resulting normalization parameter. For each SVM classifier, the classical margin-error

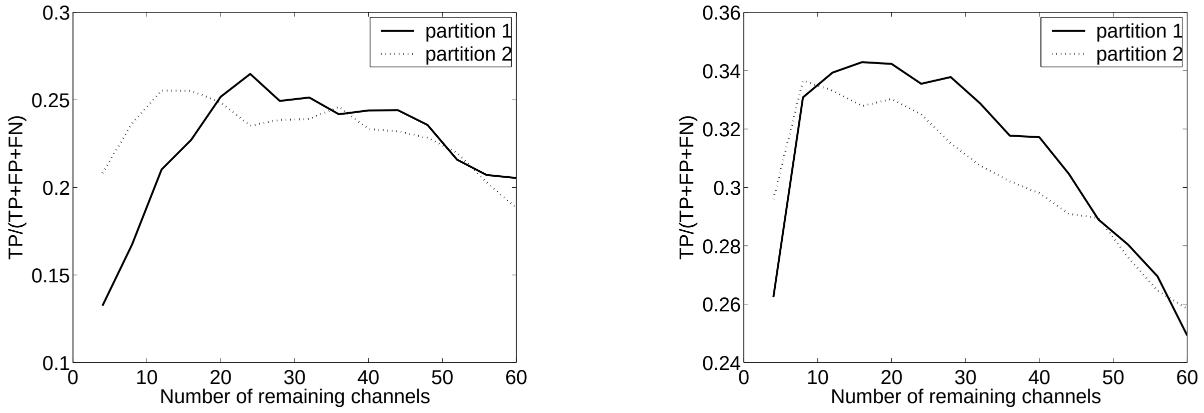

Figure 3: Evolution of the channel selection score $C_{cs}$ during channel elimination procedure for two subjects and for two different classifier trained with different training set partitions. left) Subject A. right) Subject B.

trade-off parameter $C$ and the optimal number of channels have been chosen by running the channel selection procedure for different values of $C$ and then by selecting the pair (C-number of channels) that maximizes the score $C_{cs}$. Since the validation set size is large enough (at least 6300 post-stimulus signals), we think that this strategy is sufficiently efficient especially if the $C$ values are finely sampled. In our case, we have tried $C = [0.01, 0.05, 0.1, 0.5, 1]$.

# 4 Results

This section presents the results we achieved using the above methodology. At first, some results about channel selection are presented. Then, we detail the classification performance we yielded on the test data of the competition.

## 4.1 Channel selection results

For each single classifier of decision function $f_k(x)$, channel selection has been performed based on the training set $A_k$ or $B_k$ and the related validation set. Then, we can expect that the resulting channel ranking varies from one single classifier to another.

Figure 3 gives an example of $C_{cs}$ evolution during the recursive elimination procedure for

| | Classifier index $k$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dec. Funct $f_k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Subject A | 32 | 32 | 36 | 20 | 40 | 24 | 36 | 40 | 40 | 24 | 28 | 12 | 56 | 16 | 40 | 16 | 16 |
| Subject B | 12 | 8 | 16 | 16 | 12 | 12 | 4 | 12 | 16 | 16 | 8 | 8 | 8 | 4 | 8 | 12 | 4 |

Table 1: Optimal number of channels to use in the decision function $f_k(x)$ with respects to the subject and the dataset partition $k$. For instance, $f_9$ has been trained with the dataset partition 9 and the channel selection procedure has selected 40 channels and 16 channels respectively for subject $A$ and $B$. Note that channels have been eliminated 4 by 4 in order to speed-up the elimination procedure, this explains why an optimal number of channels is always a multiple of 4.

two different dataset partitions of subject A and subject B. The plots show that there exists an optimal number of channels for each dataset and this number may vary considerably. In this figure, we can see for instance that the number of channels that maximize $C_{cs}$ can vary from 4 to 24 depending on the training set and the subject. Table 1 summarizes the number of channels that optimizes $C_{cs}$ for each decision function $f_k$ and subject. We can see that on average, for subject $A$, the relevant information on P300 are spread over more channels than for subject $B$.

Table 2 details the resulting channel ranking for subject A and B. We clearly note that rankings are variable and only few channels ($PO_7$ and $PO_8$) are consistently top ranked. We also see that some channels ($P_5$, $O_1$) are subject-dependent that is to say they have a high ranking only for one subject. Another illustration of the channel selection dependency to the subject is given in Figure 4. This figure shows the topographical histogram of the channel ranking results. It depicts how many times a given channel has been ranked in the top 12. We can remark that for subject $A$ there are only three channels that are consistently ranked whereas for subject $B$, six channels are frequently top-ranked.

These few examples of channel selection results clearly highlight the impact of the training set variability and the need of adaptivity of a P300 classification algorithm to each subject. Indeed, these results are consistent with other results in the literature [15] showing that channel

Table 2: 12 top ranked channels given without any ordering for different partitions and subject A and B .

| Data | 12 Top Ranked Channels | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | $FC_1$ | $C_2$ | $CP_3$ | $CP_z$ | $F_z$ | $F_4$ | $F_6$ | $P_5$ | $P_z$ | $P_8$ | $PO_7$ | $PO_8$ |
| A2 | $C_1$ | $CP_z$ | $CP_4$ | $AF_7$ | $AF_z$ | $F_z$ | $F_8$ | $P_5$ | $P_z$ | $PO_7$ | $PO_z$ | $PO_8$ |
| A3 | $FC_2$ | $CP_5$ | $CP_1$ | $F_1$ | $F_z$ | $FT_8$ | $T_7$ | $P_7$ | $P_5$ | $P_z$ | $PO_7$ | $PO_8$ |
| A4 | $C_3$ | $C_1$ | $F_{P1}$ | $F_2$ | $F_4$ | $F_6$ | $TP_7$ | $P_7$ | $P_5$ | $P_z$ | $PO_7$ | $PO_8$ |
| A5 | $C_z$ | $CP_5$ | $CP_2$ | $F_7$ | $F_8$ | $P_7$ | $P_z$ | $P_4$ | $P_8$ | $PO_7$ | $PO_4$ | $PO_8$ |
| B1 | $FC_5$ | $C_5$ | $C_z$ | $CP_z$ | $CP_6$ | $AF_z$ | $T_9$ | $P_1$ | $P_2$ | $PO_7$ | $PO_8$ | $O_1$ |
| B2 | $FC_1$ | $C_3$ | $C_1$ | $C_z$ | $C_4$ | $CP_3$ | $CP_z$ | $CP_4$ | $T_9$ | $P_1$ | $PO_8$ | $O_1$ |
| B3 | $C_1$ | $CP_z$ | $AF_z$ | $T_7$ | $T_9$ | $P_2$ | $P_6$ | $PO_7$ | $PO_z$ | $PO_8$ | $O_1$ | $I_z$ |
| B4 | $FC_3$ | $FC_2$ | $CP_5$ | $F_3$ | $T_9$ | $P_7$ | $P_2$ | $PO_7$ | $PO_3$ | $PO_z$ | $PO_8$ | $O_1$ |
| B5 | $FC_2$ | $C_6$ | $CP_z$ | $CP_4$ | $CP_6$ | $T_10$ | $P_3$ | $P_4$ | $PO_7$ | $PO_8$ | $O_1$ | $I_z$ |

selection should be performed on each single subject. And, even when prior knowledge about the mental task to be performed by the subject are available, channel selection allows the system to adapt to the mental activity of the subject, leading then to better performances [9].

## 4.2 Results on the test set of BCI III competition

Test sets have been processed similarly to the training set and then are fed to our ensemble of classifiers. For the competition, performances have been evaluated based on the correctness of predicted characters in the test sets. Hence, we are considering the 36-class classification problem, and each predicted character has been obtained according to the method described in section 3.2. Table 3 depicts the performance results we achieved on the test sets with respect to the number of sequences.

Based on the competition evaluation criteria, we achieve a correct classification performance of 73.5% and 96.5% for respectively 5 and 15 sequences. Remember that this performance has been evaluated on a test set composed of 200 spelling characters.

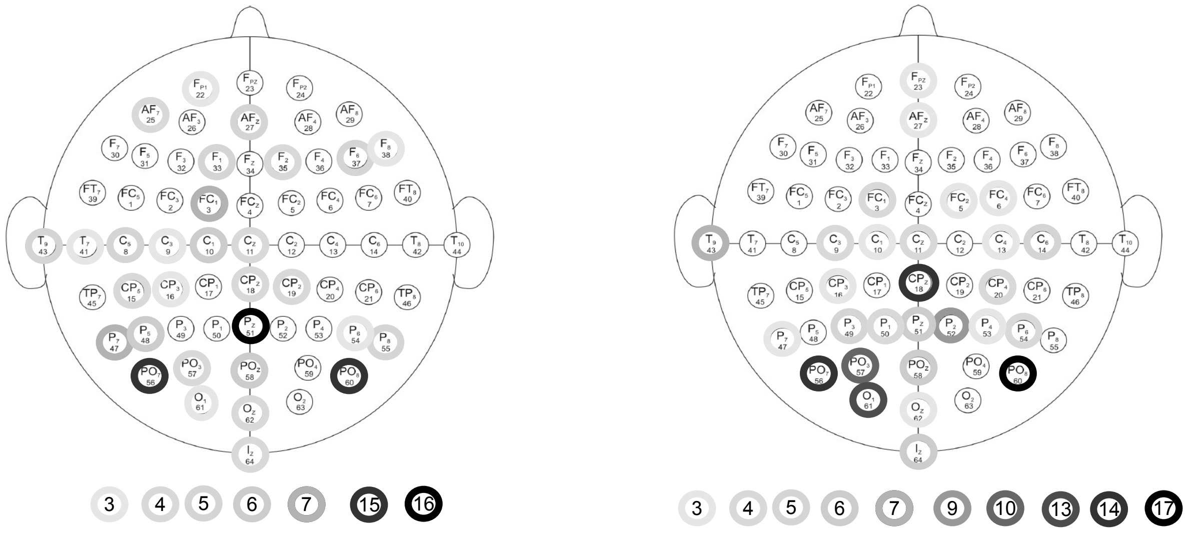

Figure 4: Topographical histogram of channels ranking for (left) Subject A. (right) Subject B. Each channel has been contoured with a circle. The gray scale denotes the number of times, given in each circle below the scalp, a channel has been ranked in the top 12. For instance, for subject A, channel $P_z$ has been ranked 16 times out of the 17 classifiers in the top 12 channels. For subject $B$, channel $P_2$ has been ranked 10 of the 17 classifiers in the top 12 channels.

The main advantage of a classification algorithm competition is to provide an unbiased comparison of different algorithm performances on the same dataset. Hence, we present on Table 4 the results of other competitor algorithms. These results have been obtained from the BCI competition website [1].

Interestingly, the runner-up competitors have proposed an algorithm similar to ours. Indeed, they use a multiple classifier strategy where each single classifier is a SVM. However, instead of summing the score of each single classifier, they use a voting strategy. We believe that this is a key difference since using the sum instead of the vote keeps trace of the confidence of one classifier in the score $S_{r|c}$ and thus in the final decision. Their algorithm also differs in the channel selection procedure. They have decided to use a fixed number of channels, chosen in a ad-hoc way. These channels are different for each subject. Although somewhat similar, their algorithm and ours provide significantly different performances especially when only 5 sequences are considered (up to 18%). The other competitors use different approaches which

Table 3: Classification performance in % of correctly recognized characters for the 2 subjects and for increasing number of sequences.

| | Nb of sequences | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subject | 1 | 2 | 3 | 4 | 5 | 10 | 13 | 15 |
| A | 16 | 32 | 52 | 60 | 72 | 83 | 94 | 97 |
| B | 35 | 53 | 62 | 68 | 75 | 91 | 96 | 96 |
| Mean | 25.5 | 42.5 | 57.0 | 64.0 | **73.5** | 87.0 | 95.0 | **96.5** |

Table 4: Average classification performance in % of the 3 best algorithms of the competition.

| | Nb of sequences | |
|---|---|---|
| Algorithms | 5 | 15 |
| Our algorithm | **73.5** | **96.5** |
| 2nd ranked algorithm | 55.0 | 90.5 |
| 3rd ranked algorithm | 59.5 | 90 |

lead to performances of about 55% and 90% recognition rate for 5 and 15 sequences.

### 4.3 Analysis and improvements of this winning strategy

The approach we proposed has been the winning strategy. Here, we present a short analysis on the reasons of such good performances.

As we stated in previous sections, two issues arise in this classification problem. Because of the low signal-to-noise ratio of the signals, we have proposed an averaging strategy. Ensemble averaging is a common method for enhancing event-related potentials signal-to-noise ratio. In our classification method, two forms of averaging are performed, one with respects to the sequences, on the other one with respects to the different classifier scores. The averaging according to sequences is usual for BCI P300 speller and has already been proposed by Kaper et al. [8, 10]. Such an averaging is one of the reasons why character recognition performance increases with

Table 5: Comparison of the performances of some SVM composing 3 ensembles : using channel selection, using all 64 channels and using 8 prefixed channels. Each SVM has been trained on a particular partition of the dataset. The results we report are the performance of such a classifier evaluated on the test set and after 5 and 15 sequences. We can see that our channel selection procedure allows each single SVM to achieve consistently better performances than using either all or 8 channels.

| | Performances after 5 and 15 sequences | | | | | |
|---|---|---|---|---|---|---|
| Dataset | Optimal channels | | 64 channels | | 8 channels | |
| $A_1$ | 26 | 66 | 22 | 55 | 24 | 60 |
| $A_2$ | 41 | 69 | 22 | 61 | 15 | 54 |
| $A_3$ | 28 | 64 | 19 | 59 | 5 | 27 |
| $A_4$ | 36 | 81 | 24 | 56 | 17 | 53 |
| $A_5$ | 39 | 75 | 27 | 69 | 23 | 52 |
| $B_1$ | 62 | 93 | 52 | 80 | 41 | 76 |
| $B_2$ | 61 | 90 | 49 | 73 | 31 | 54 |
| $B_3$ | 56 | 81 | 45 | 65 | 36 | 65 |
| $B_4$ | 57 | 89 | 49 | 81 | 47 | 65 |
| $B_5$ | 53 | 89 | 59 | 88 | 33 | 70 |

the number of sequences. From our point of view, the most important contribution of our classification approach is the ensemble classifier averaging. Interestingly several other competitors have also proposed a classification method based on ensemble averaging classifiers. And we think that this is a promising strategy.

We carried out some further analysis after the competition in order to get some more insights on our method. For instance, we have measured the performance of the ensemble of SVM without any channel selection, and using only 8 prefixed channels ($F_z$, $C_z$, $P_z$, $C_3$, $C_4$, $P_3$, $P_4$, $O_z$) . These eight channels are among the ten ones used by Meinicke et al. [10]. For these ensembles of classifiers, the regularization parameters have been set by the same validation procedure as described in section 3.4.

At first, we have compared in Table 5 the performance of some SVMs composing each ensemble. We can see that using our channel selection procedure yields each single SVM to better performance. The gain of performance can be of the order 20% for subject $A$ and 15% for subject $B$. This suggests that for each SVM trained on a small part of the training set, channel selection should be carried out.

Performance results concerning these ensembles of SVM are given in Table 6. We see from this table that without channel selection we achieve similar performance than with channel selection. Hence, we can remark that the loss of performance of each SVM of the ensemble when using 64 channels (see Table 5), has been compensated by the ensemble strategy, either owing to the larger amount of data or owing to the output score averaging. However, we also see that using a prefixed choice of channels leads to worse performance : the ensemble strategy does not compensate for the loss of information due to the limited number of channels.

All these results suggest that, for this competition, one should have either used all channels or performed channel selection. In this context, our channel selection procedure only acts as a subject-dependent dimensionality reduction. Other experiments we carried out on BCI competition 2003 datasets [2] and reported in Rakotomamonjy et al.[12] have also shown that the channel selection procedure only slightly enhances performances .

Furthermore, we have tried to empirically analyze the contribution of the ensemble of SVM compared to a single SVM trained on all examples. Performances of single SVM are also reported on Table 6. For these results, we have tried different regularization parameters and reported the best achieved performances on the test set. We see that when using all channels, a single SVM gives similar results than our algorithm for 15 sequences. However, for only 5 sequences, ensemble of SVM performs better, of the order of 5%, than a single SVM. Although disappointing from the competition point of view , we think that for a real application perspective where high classification rate using few sequences are desired, this is an interesting advantage. Again, results on BCI competition 2003 datasets and reported in Rakotomamonjy et al.[12] have also shown that differences between single and ensemble of SVM are more significant when considering only a small number of sequences. We can also see that when only 8 fixed channels are used, the

Table 6: Table of performances of our algorithm and other related algorithms. For instance, we have the classification performance in % of an ensemble of SVM and a single using all 64 channels.

| | Nb of sequences | |
|---|---|---|
| Algorithms | 5 | 15 |
| Our algorithm | **73.5** | **96.5** |
| Ensemble SVM without channel selection | 74.5 | 95.5 |
| Ensemble SVM with 8 prefixed channels | 40.0 | 80.0 |
| Single SVM without channel selection, C=0.01 | 69.5 | 96.5 |
| Single SVM with 8 prefixed channels , C=1e-5 | 31.0 | 70.0 |

ensemble of SVM gives far better results than a single SVM.

A similarity between our ensemble of SVM and a single SVM can also be noted. Indeed, we know that a linear combination of linear classifiers (see equation 3 ) is just another linear classifier. Explicitly, equation 3, which gives the score assigned by our ensemble of SVM to a row or a column can be written as :

$$S_{r|c} = \sum_i y_i \tilde{\alpha}_i \left\langle \frac{1}{J} \sum_{j=1}^{J} x_{r|c}^{(j)}, x_i \right\rangle + \tilde{b} \tag{4}$$

with

$$\tilde{\alpha}_i = \begin{cases} \left( \frac{1}{K} \sum_{k=1} \alpha_i^{(k)} \right) & \text{if } i \in P_k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \tilde{b} = \frac{1}{K} \sum_{k=1} b^{(k)}$$

Hence, we could have trained a single linear classifier with a particular, but unknown loss and regularization functions and get the same classifier given by an ensemble of SVM.

From this similarity, we can understand that the main difference between our approach and a single SVM (with the usual hinge loss) is the values of $\tilde{\alpha}_i$ and $\tilde{b}$. In our ensemble of SVM, the regularization parameter $C$ of each SVM is selected by cross-validation and thus they may be different from one SVM to another. From a single SVM point of view, this means that each training example $x_i$ should have its own regularization parameter $C_i$. Typically, in our case, training examples coming from the same partition would have the same regularization parameter. This implies that misclassified training examples (bounded support vectors) would get different values of $\alpha_i$ (which upper bound depends on $C_i$). Hence, compared to a single SVM with a single parameter $C$, our ensemble of SVM differently weights misclassified support vectors. Besides, since the support vector numbers are about 1/3 of the training examples for both strategies, we can think that the main difference between these two approaches is these adaptive weights given to bounded support vectors.

The algorithm described above is efficient but we think there is still room for improvements. For instance, the following points can be investigated:

- the idea of using an ensemble of classifiers has been the keystone of our algorithm. However, this idea relies on the dataset partitioning and thus on how the training set has been clustered. Then, it would be interesting to investigate the importance of such a clustering. For instance, the question that may arise is the following: is it better to train a classifier on data from a small part of the data space or on data sampling all data space?

- the problem of inter-subject variability has not been addressed by our algorithm since we have only used signals from the same subject for learning and testing. This issue of inter-subject learning is important in order to make this BCI speller efficient with a new patient and without the need of a training session. For instance, using the ensemble of classifiers trained with EEG signals from subject A for classifying the test set of subject B yields to a performance rate of only 26%.

## 5 Conclusion

This paper has presented the algorithm that achieved the best performance for the dataset II of the BCI competition III 2005 [1]. The novelty in the approach is that the data sets have been split in several partitions and a classifier has been trained on each of this partition. The outputs of all classifiers are finally summed up to get a final decision. Results we achieved with this classifier are rather good if we consider only 5 sequences.

We are aware that results presented in this paper are somewhat limited because of the small number of subjects and the small number of testing sets. However, the presented results have been validated and compared in an unbiased way by the competition. Thus, it is very likely that the methodology presented here should also perform well within an online BCI environment. Our next research goes towards this direction. We plan to contact the dataset provider and to perform a large-scale analysis of this algorithm using a larger amount of data.

Furthermore, if we want to enhance the bit rate transfer of this BCI P300 speller more works have to be carried out. Our next objective is now to consistently achieve a recognition rate of up to 90% with only 5 sequences and to efficiently address the intra-subject variability. This is the subject of our present research.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors views.

## Appendix

### Validation sets

For computational reasons, we have used for model selection a particular splits of the training data. Remember that each spelled character generates $180 = 12 \times 15$ post-stimulus signals. With all the 85 characters, we have built 17 partitions of 5 spelled characters, hence of 900 signals. Then for the model selection procedure, we have used the following splits. When a SVM is trained on a partition $k$ with $k \in [1, \cdots, 8]$, then its validation set is all partitions from $[1, \cdots, 8]$ except $k$. A similar procedure is performed for a SVM trained with a partition $k \in [9, \cdots, 17]$. Hence, the validation set is composed of 6300 or 7200 post-stimulus signals. We have considered that such an amount of data is sufficiently large for evaluating accurately each SVM model.

# References

[1] B. Blankertz, "Bci competition III webpage." [Online]. Available: http://ida.first.fraunhofer.de/projects/bci/competition_iii

[2] B. Blankertz, K.-R. Mueller, G. Curio, T. Vaughan, G. Schalk, J. Wolpaw, A. Schloegl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroeder, and N. Birbaumer, "the BCI competition 2003: Progress and perspectives in detection and discrimination of eeg single trials," *IEEE Trans. Biomed. Eng*, vol. 51, no. 6, pp. 1044–1051, 2004.

[3] B. Blankertz, K.-R. Mueller, D. Krusienski, G. Schalk, J. Wolpaw, A. Schloegl, G. Pfurtscheller, J. del R. Millan, M. Schroeder, and N. Birbaumer, "The BCI competition iii: Validating alternative approaches to actual bci problems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 153–159, 2006.

[4] E. Donchin, K. Spence, and R. Wijeshinge, "The mental prosthesis : assessing the speed of p300-based brain-computer interface." *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 174–179, 2000.

[5] L. Farwell and E. Donchin, "Talking off the top of your head : toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[6] Y. Grandvalet, "Bagging equalizes influence," *Machine Learning*, vol. 55, no. 3, pp. 251–270, 2004.

[7] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[8] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter, "BCI competition 2003 - dataset IIb: Support vector machines for the P300 speller paradigm," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1073– 1076, 2004.

[9] T. Lal, M. Schroeder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schoelkopf:, "Support vector channel selection in BCI," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1003–1010, 2004.

[10] P. Meinicke, M. Kaper, F. Hoppe, M. Heumann, and H. Ritter, "Improving transfer rates in brain computer interfacing : A case study." in *Advances in Neural Information Processing Systems 15*, S. T. In S. Becker and e. K. Obermayer, Eds., vol. 15, 2003, pp. 1107–1114.

[11] A. Rakotomamonjy, "Variable selection using SVM-based criteria," *Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.

[12] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado, "Ensemble of SVMs for improving brain-computer interface p300 speller performances," in *15th International Conference on Artificial Neural Networks*, 2005.

[13] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000 : a general-purpose brain-computer interface (BCI) system," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[14] B. Schoelkopf and A. Smola, *Learning with Kernels.* MIT Press, 2001.

[15] M. Schroeder, T. N. Lal, T. Hinterberger, M. Bogdan, J. Hill, N. Birbaumer, W. Rosenstiel, and B. Schoelkopf, "Robust eeg channel selection across subjects for brain computer interfaces." *EURASIP Journal on Applied Signal Processing*, vol. 19, pp. 3103–3112, 2005.

[16] H. Serby, E. Yom-Tov, and G. F. Inbar, "An improved p300-based brain-computer interface," *IEEE Trans Neural Syst Rehabil Eng*, vol. 13, no. 1, pp. 89–98, 2005.

[17] M. Thulasidas, C. Guan, and J. Wu, "Robust classification of eeg signal for brain-computer interface," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 14, pp. 24–29, 2006.