ABSTRACT

This paper evaluates an algorithm based on Support Vector Machines to analyze EEG data from the P300 speller Brain-Computer Interface paradigm. We evaluated the performance of this technique on own experimental data from 8 subjects and achieved high transfer rates of up to 97.57 bits/min (mean 47.26 bits/min) within subjects. We then investigated how well the classifier generalizes when it is trained on data from a set of several subjects and then applied on data from a new subject to use this BCI in a pretrained fashion. Transfer rates up to 61.04 bits/min were achieved (mean 17.64 bits/min) for this situation indicating an encouraging generalization performance.

1. INTRODUCTION

Brain-Computer Interfaces (BCI) enable people to steer computers merely by thoughts [2]. This could be helpful especially for highly paralyzed persons suffering from e.g. Amyotrophic Lateral Sclerosis (ALS) [3]. EEGs are well-suited for this problem since they have a high temporal resolution and are non-invasive. In this study we utilize the Machine Learning technique of Support Vector Machines (SVM) to analyze data from the P300 speller paradigm in order to obtain high speed in terms of transferred information (bits) per minute [4, 5]. Furthermore, we investigate how well this algorithm generalizes among subjects. A good generalization performance would allow using a pretrained Brain-Computer Interface without any adaptation of the algorithm to the specific subject. Initial investigations have been made by the authors in a previous study but were restricted to data from a slow variant of the paradigm [6].

Fig. 1. Stimulus matrix from the P300 speller paradigm with one column highlighted.

This work was supported by the project SFB 360 and Grant Ne366/4-1 from the Deutsche Forschungsgemeinschaft. We thank Florian Hoppe and Peter Meinicke for cooperation on constituent work, and the Dep. of Cognitive Psychology and Werner Klotz for experimental support. We have used the SVM algorithm from the libsvm toolbox by Chang & Lin [1].

2. P300 SPELLER PARADIGM

The oddball paradigm teaches that rare attended stimuli are accompanied by a positive deflection in the EEG after about 300ms. This so-called P300 component is exposed by nearly every human being and is therefore independent of training of participants. The P300 speller paradigm utilizes this component for BCI as follows: The rows and columns of a 6 × 6 matrix containing 36 symbols are flashing sequentially (see Fig. 1). The subject is instructed to concentrate on one symbol in this matrix by counting how often it flashes. Thus, according to the oddball-paradigm, a P300 should result when the row/column with the attended symbol flashes. Therefore, by identifying this P300 component in the EEG pattern it is possible to infer the attended symbol [7, 8].

3. SVM CLASSIFICATION

Support Vector Machines, a Machine Learning approach based on structural risk minimization [9] have been shown to perform binary classification with very good generalization.

Fig. 2. Support Vector Machines find the optimal hyperplane (solid line) to separate two classes by maximizing the margin γ. It can be described by the vector w and the bias term b.

To perform such a classification, in the simplest case a hyperplane (described by the vector w and the bias b) can be constructed which separates the two classes from each other as depicted in Fig. 2. A Machine Learning classifier would need to find such a hyperplane from l given examples of training vectors $x_i$ with corresponding labels $y_i$: $(x_i, y_i)$, ..., $(x_l, y_l) \in \mathbb{R}^n \times \{-1, 1\}$. Projecting a new vector $x$ from a disjoint test set on this vector $w$ would reveal its class by the sign of the projection:

$$f(x) = w \cdot x + b. \quad (1)$$

Support Vector Machines maximize the margin $\gamma$ (see Fig. 2) in order to find the optimal hyperplane with the best generalization.
The Lagrangian dual of this optimization task is

\[
\min \frac{1}{2}||w||^2 + C \sum_i \xi_i \\
\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i > 0 \ \forall i.
\]

The Lagrangian dual of this optimization task is

\[
\max \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\
\text{s.t.} \quad 0 \leq \alpha_i \leq C \ \forall i.
\]

Solving this problem yields \( w = \sum_i^{N_s} y_i \alpha_i x_i \), with \( N_s \) as the number of resulting support vectors. This results for (1) in

\[
f(x) = \sum_i^{N_s} y_i \alpha_i (x \cdot x_i) + b.
\]

In this study, a Gaussian Kernel \( K(x, x_i) = \exp \left(-\frac{||x-x_i||^2}{\sigma^2}\right) \) maps the data space to a higher dimensional feature space making separation much more likely for nonlinear cases. This yields the nonlinear discriminant function

\[
S(x) = \sum_i^{N_s} y_i \alpha_i K(x, x_i) + b.
\]

The SVM classifier is controlled by the regularization parameter \( C \) and the Kernel bandwidth \( \sigma \). In this study, we choose them by crossvalidation (cf. next section).

4. CROSSVALIDATION

It is crucial to have disjoint sets for training and testing to avoid overfitting of a Machine Learning classifier. Crossvalidation divides a dataset into \( k \) subsets of data for a \( k \)-fold crossvalidation and takes \( k - 1 \) subsets for training and the omitted subset for testing [11]. The sets are permuted and the average rate of the \( k \) test set classifications is taken as a measure for classification rate.

<table>
<thead>
<tr>
<th>outer</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>inner</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Example of division into subsets for the outer (upper row) and inner crossvalidation (lower row) with \( k=5 \) and \( m=6 \) subsets. 'A' refers to training subsets, and 'B' to the test set. The position of 'B' is permuted systematically.

Beside this outer crossvalidation, an inner crossvalidation can be performed to find optimal values for the classifier parameters \( C \) and \( \sigma \). Here, the \( k - 1 \) sets are further divided into \( m \) sets (Fig. 3). Different values for the classifier parameters are chosen to train data from \( m - 1 \) sets, and the omitted set is then classified. Afterwards, the \( k - 1 \) sets from the outer crossvalidation are trained with these parameters.

5. EXPERIMENTAL SETUP

Eight volunteers attended (age 20-34). Each subject was instructed to count the flashings of one symbol in the matrix. This symbol was randomly chosen by the presentation program and presented to the subject in advance of each trial. A trial consisted of 4 to 6 repetitions of 12 flashes (data according to one flash is called 'epoch'). Four subjects performed 720 trials each, four subjects performed 450 trials each. The interstimulus interval (ISI) was 140ms which results in an overlap of epochs (remember, the P300 onset is expected after about 300ms).

5.1. Data acquisition

According to the international 10-20 system, Ag/AgCl electrodes were applied at Fz, Cz, Pz, Oz, C3, C4, P3, P4, PO7, and PO8 and recorded by a Neuroscan Synamps amplifier. Data was sampled with 200Hz and analyzed offline.

5.2. Preprocessing

Time series of 800ms EEG amplitude values after flashing a row or column were extracted and bandpass filtered (0.5-9Hz). Data from all 10 electrodes was concatenated and the resulting vector was then normalized for each trial to an interval of [-1,1].

5.3. Classification algorithm

Throughout this study, inner crossvalidation was done on equal sets of positive and randomly chosen negative samples for the P300. In outer crossvalidation, the classifier was trained on the data from the inner set with the optimal parameters and then applied on the data of the test set. The test set consists of complete sequences of each 2 positive and 10 negative examples.

To combine data from several flashing sequences, the values of the discriminant function (5) from corresponding rows/columns from the sequences were accumulated: \( S(x_{ik}^{row}) = \sum_{s=1}^{s_{row}} S(x_{ik}^{srow}) \). \( S(x_{ik}^{row}) \) reflects the score of the epoch \( x_{ik}^{srow} \) from the \( i \)-th row of the \( k \)-th trial. After \( n \) trials, the row with the highest value was chosen to represent the row with the P300.

6. CLASSIFIER ADAPTATION WITHIN SUBJECTS

6.1. Data analysis

A 5-fold outer crossvalidation was performed. To keep computational costs low we only used data from one of the remaining 4 sets to train the classifier. The inner crossvalidation was also 5-folded and consisted of randomly chosen 70 samples for each subset.

6.2. Results

The P300 classification rates were taken as the average from inner crossvalidation for optimal parameter values. These rates vary between 0.73 (subject 2) and 0.91 (subject 3) as depicted in Fig. 4. Mean P300 classification was 0.79 (SD=0.07). Thus, a single EEG-Trial can be divided into P300 and non-P300 Trials with 79% accuracy on the average.

But to infer a symbol from P300 classification, 2 x 6 classifications need to be combined and exactly one row and one column has to be selected to contain a P300 to identify a symbol. Fig. 5 shows the results for this part. Classification rates for the
different timesteps, which correspond to repetitions of 12 flashes are depicted. After 1.68 seconds, rates range between 0.25 and 0.7. From this point in time, the mean rates increase from 0.44 (SD=0.15) up to 0.9 (SD=0.09) after 8.4 s.

6.3. Calculating bit rates

For better comparison between BCI approaches, a common technique to assess speed is to calculate the bits of information provided within a minute (bits/min). This bit rate is calculated via

\[ B_t = \frac{60}{t} \left( \log_2 N + p \log_2 p + (1-p) \log_2 \left( \frac{1-p}{N-1} \right) \right). \]

\( N \) is the number of available symbols, \( p \) denotes the probability of a correct choice and \( t \) the time for a choice. Fig. 6 shows that the transfer rates decrease in the average with more repetition steps, although it sometimes increases for some subjects from step 1 to step 2. Thus, although classification accuracy gets better with more repetitions, mean transfer rates do not improve since every repetition takes its time. The increase in accuracy does not compensate for longer duration in time. For the first step, bit rates vary between 18.41 bits/min and 97.57 bits/min with a mean of 47.26 bits/min (SD=26.38 bits/min).

7. GENERALIZING AMONG SUBJECTS

In the previous section, the classifier was trained and evaluated on data from the same subject. But it would be very useful to be able to use a trained classifier so that it would not be necessary to first adapt the system to a new user. This requires information about the generalization capabilities of the classifier between subjects.

7.1. Data analysis

Crossvalidation was in principle done as before, but for the test set in the outer crossvalidation, data from one subject was used for classification, and data from the other subjects was taken together to train the classifier. This procedure gives information about how well the classifier would perform on data from a ‘new’ subject it has never seen before. To keep computational costs low, we only took 100 examples for each subject for the test set in the outer crossvalidation. In the inner crossvalidation, 50 samples (an equal number of positive and negative samples, chosen randomly) were
taken from each of the remaining 7 subjects, resulting in 350 samples and evaluated within a 7-fold inner crossvalidation scheme.

7.2. Results

P300 classification rates vary between 0.68 and 0.72 with a mean of 0.70 (SD= 0.02) as depicted in Fig. 7. When calculating rates for symbol inference, results between subjects vary significantly (cf. Fig. 8), and the rates vary between 0.08 and 0.52 after 1.68 s for the different subjects. In the mean, classification accuracies increase with timesteps from 0.24 (SD=0.15) after 1.68 s to 0.64 (SD=0.24) after 8.4 s.

Fig. 8. Classification rates for symbol prediction when taking data from other subjects for training than for testing. The numbers in the legend refer to the different subjects. Mean rates are drawn bold.

Bit rates show that a trained classifier can work very well for some subjects, while it lacks performance in others. This difference between subjects reflects similar differences as in the previous section. Well performing subjects stay good and bad performing subjects stay bad. The best subject achieved the rather high bit rate of 61.04 bits/min. The clear trend from the previous section, that the highest bits/min are achieved for just a single sequence of 12 flashes is not true anymore for the mean where highest bit rates are achieved for 2 repetitions (17.64 bits/min; SD=17.56 bits/min).

Fig. 9. Bit rates in symbol inference calculated for a new subject in the test set. Mean bit rates are drawn bold.

8. CONCLUSION

Very high transfer rates up to 97.57 bits/min with a mean of 47.26 bits/min were achieved within subjects. This is the highest bit rate for EEG-based BCIs the authors are aware of.

A classifier which was trained on data from a set of subjects and applied on data from a new subject, achieves a significantly lower performance (17.64 bits/min in the mean) than in the within-condition. However, some subjects performed very well and achieved up to 61.04 bits/min.

Thus, this approach has interesting potential for a pretrained system. A subject could use this system and achieve more than 17 bits/min in the average without any prior acquisition of data from this subject. However, higher transfer rates might be possible with more computational costs in the classifier training (i.e. giving it more examples) and a more elaborated preprocessing. Furthermore, it could be useful to find more coherent prototypical subjects to train the classifier to be able to achieve better performance in the average.

9. REFERENCES


