

A Comparison Study on Machine Learning Algorithms Utilized in P300-based BCI

Rami J Oweis*, Naser Hamdi, Adham Ghazali and Khaldoun Lwissy

Biomedical Engineering Department, Faculty of Engineering, Jordan University of Science and Technology, P.O. Box 3030, Irbid 22110, Jordan

Abstract

This study addresses Brain-Computer Interface (BCI) systems meant to permit communication for those who are severely locked-in. The current study attempts to evaluate and compare the efficiency of different translating algorithms. The setup used in this study detects the elicited P300 evoked potential in response to six different stimuli. Performance is evaluated in terms of error rates, bit-rates and runtimes for four different translating algorithms; Bayesian Linear Discriminant Analysis (BLDA), Linear Discriminant Analysis (LDA), Perceptron Batch (PB), and nonlinear Support Vector Machines (SVMs) were used to train the classifier whilst an N-fold cross validation procedure was used to test each algorithm. A communication channel based on Electroencephalography (EEG) is made possible using various machine learning algorithms and advanced pattern recognition techniques. All algorithms converged to 100% accuracy for seven of the eight subjects. While all methods obtained fairly good results, BLDA and PB were superior in terms of runtimes, where the average runtimes for BLDA and PB were 13 ± 2 and 15.6 ± 6 seconds, respectively. In terms of bit-rates, BLDA obtained the highest average value (22 ± 12 bits/minute), where the average bit-rate for all subjects, all sessions, and all algorithms was 18.76 ± 10 bits/minute.

Keywords: Brain-Computer interface; P300; Evoked potential; Linear discriminant analysis; Support vector machines

Introduction

The electrical activity of the brain i.e., electroencephalography (EEG) originates mainly from the cerebral cortex. The amplitude of the EEG signal is commonly in the range of 0 to $\pm 100 \mu\text{V}$, and its frequency is in the range of 0 to 100 Hz. The four different EEG waves, namely, alpha, beta, theta, and delta, are characterized by its own unique properties. Although not observed separately, one property is usually dominant over the others depending on the state. One of the problems associated with the EEG frequency range is the signal susceptibility to ambient noise. Noise sources such as the 50 Hz power line interference and impedance fluctuations tend to complicate and hinder the acquisition of a strong EEG signal with a high Signal to Noise Ratio (SNR) in addition to cause motion artifacts [1].

Since its appearance in 1923, EEG has been used mainly as a diagnostic tool for neurological disorders. Recently, there has been growing interest in using EEG as a control channel to help people suffering from a health condition apparently similar to coma in which patients are mute and totally paralyzed, except for eye movements, but stay conscious. This condition usually results from massive hemorrhage, thrombosis, or other damage, affecting upper part of brain-stem, which destroys almost all motor function, but leaves the higher mental functions intact. People having such a condition are identified as locked-in. Those people produce the same spatiotemporal activation patterns on intention to use an extremity as those observed in healthy individuals [2-4].

In order to extract useful information from the EEG data, pattern recognition algorithms may be employed. There are different techniques for pattern recognition necessitating the careful selection and design of an appropriate method to a specific problem. Most schemes are based on statistical probability evaluation [5], neural networks [6], support vector machines (SVMs) [7], or other similar techniques.

A functional and seamless human-machine interface will have a profound impact on those suffering from neurological disorders that make them unable to communicate or manipulate their surroundings.

While some people with limited handicaps may achieve this communication through physical contact with a machine—such as using a keyboard, manipulating a joystick, or even issuing voice commands, others are severely handicapped and unable to communicate through the normal neuromuscular channels, and may benefit from a special interface. A Brain Computer Interface (BCI) is a system that allows communication with the central nervous system (CNS) by translating brain signals into commands a machine is able to understand [6,8-10]. Most BCIs are special types of Human Machine Interfaces (HMIs), which are characterized by the use of Electroencephalography (EEG) as the main channel of communication. The use of EEG for communication is an important feature of BCIs allowing for a communication pathway independent of patterns produced by motor activity, where the command signal is extracted directly from cortical brain activity and is independent of efferent neuromuscular channel activity.

There are currently several major categories of BCIs in use that are classified based on the type of neurophysiologic signal they utilize. These categories include, but are not limited to, Visual Evoked Potentials (VEPs), P300 elicitation, alpha and beta rhythm activity, slow cortical potentials (SCPs), and microelectrode cortical neuronal recordings [2,3,11]. The P300 waves are evoked potentials that are elicited in response to specific stimuli, while SCPs occupy the lowest frequency range of the EEG signal and are associated with cortical activation and deactivation. In the SCPs, negative shifts correspond to cortical activation and positive shifts correspond to cortical deactivation. The alpha (or Mu rhythms) and beta activities cover the frequency ranges

*Corresponding author: Rami J Oweis, Biomedical Engineering Department, Faculty of Engineering, Jordan University of Science and Technology, P.O. Box 3030, Irbid 22110, Jordan, Tel: 0096227201000; Fax: 0096227201074; E-mail: oweis@just.edu.jo

Received May 18, 2013; Accepted June 15, 2013; Published June 19, 2013

Citation: Oweis RJ, Hamdi N, Ghazali A, Lwissy K (2013) A Comparison Study on Machine Learning Algorithms Utilized in P300-based BCI. J Health Med Informat 4: 126. doi:10.4172/2157-7420.1000126

Copyright: © 2013 Oweis RJ, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

8–12 Hz and 12–30 Hz, respectively. These are thought to be activities of the sensory/motor cortex [12]. As for the microelectrode cortical neuronal recordings, it is an invasive technique involving the direct contact of microelectrodes with brain tissue. Most BCI systems aim to distinguish between different signals based on subject intention. A P300-based BCI system emits different commands based on the time at which the P300 is elicited, while SCP-based systems detect commands based on positive or negative voltage shifts [3,13]. Furthermore, P300 is an event-related potential that can be seen as a positive deflection of the normal EEG in response to stimuli after approximate 300 ms latency [14]. The algorithm was first introduced by MacKay [15] in Bayesian interpolation, and later implemented for P300 wave detection by Hoffmann et al. [16]. It exhibits viability for online BCI systems due to the recursive computation of the hyper-parameters.

Various techniques are applied to classify data and features. For example, LDA assumes linear separation between different states, and in order to ensure separation, a simple mathematical procedure is applied [17]. On the other hand, Principal Component Analysis (PCA), which is a statistical method that searches for components of high significance in representing the data while eliminating those of low contribution, can be seen as a search for directions that are efficient for representing the data [5]. Independent Component Analysis (ICA), sometimes referred to as “blind source separation”, is a statistical procedure used to separate signals that are mixed linearly and randomly, assuming that these signals originate from independent sources [18-20]. BLDA is an iterative procedure, which aims to compute the posterior probability using hyper-parameters [15]. ICA and PCA were not investigated in this study due to the additional computational complexity introduced into the problem, which is contrary to our goal of providing a simple and efficient online EEG processing tool.

Sellers and Donchin [21] developed a P300-based BCI using a four choice paradigm (Yes, No, Pass, End), where classification was based on Stepwise Discriminant Analysis (SWDA). The aim of their study was to determine whether Amyotrophic Lateral Sclerosis (ALS) patients could use P300 BCIs as an alternative communication channel. Using the Berlin brain computer interface (BBCI) [2], data was collected based on stimuli that evoked readiness potentials, and then preprocessed using fast Fourier transform (FFT) filters. Subsequently, Fisher Linear Discriminant Analysis (FLDA) was trained to classify the data. The authors obtained good results, with an error approaching zero within 500 ms, and a bit rate of 37 bits/min for a spelling task.

In order to conduct a comparison of various algorithms, different electrode configurations, and the consequent bit rates, a P300 based BCI was developed by Hoffmann et al, providing an evaluation of two classification algorithms, specifically, FLDA and BLDA [16]. The study suggested that BLDA obtained a higher bit-rate and classification accuracy. Furthermore, the accuracy and speed (bit-rate) increased proportionally with the dimension of the data. After an extensive first session of supervised algorithm learning and feedback, a study based on the datasets provided by BCI competition 3 incorporated the use of adaptive linear discriminant analysis (ALDA) for classification of different motor imageries [22]. The study showed that ALDA outperformed LDA during supervised learning sessions with higher decoding power over time. A real-time independent BCI system was implemented in the Graz-BCI. The system aimed at distinguishing the different motor imageries based on Event-related De-synchronization (ERD) and Event-related Synchronization (ERS) of the Mu and beta rhythms [12].

The present work introduces an optimized P300-based BCI system.

Detecting the emergence of the P300 wave in a time sequence is not easy due to variability among subjects in amplitude, latency and duration. Thus instead of designing subject-based detection algorithms, this study utilized machine learning techniques in which a classifier is trained to detect target signals from a training set. It also attempts to evaluate the performance of four different translating algorithms (BLDA, LDA, PB, and Nonlinear SVM).

Materials and Methods

This study employed the same data set used in the investigation by Hoffmann et al. [16], which comprised four healthy subjects and four subjects with neurological deficits. The recorded EEG data were based on visual stimuli (TV, telephone, lamp, door, window, and a radio) that evoked the P300 component. Each subject recorded four sessions, one minute for each class for six different classes, giving a total of 24 minutes of recording. Subjects were asked to focus on a specific image for each run; while the sequence of stimuli was randomly presented. Several performance measures were computed to develop a comparison between the different methods. The ultimate objective is to determine which of the methods would be most suitable for accurate real-time communication. The study employs Bayesian Linear Discriminant Analysis (BLDA) to detect the emergence of the P300 wave in the time series. It utilizes a set of different algorithms, including Linear Discriminant Analysis (LDA) and perceptron neural network programming, in order to train a linear classifier. In addition, a nonlinear algorithm (nonlinear SVM) is also used to train a nonlinear classifier for the sake of performance comparison with the other linear algorithms.

The BCI system was designed for real-time analysis relying on prior training of the classifier. The online data processing was developed using Matlab and Simulink (Math Works, Inc., USA). During acquisition, data are processed online sample by sample. Recording starts at the initialization of stimuli. A timing function specifies the feature vector lengths and time points in reference to the start time of recording. Thus the predefined timing function, based on subject intention and flags, emitted by the different stimuli can produce the six different class labels online during the training phase of the classifier.

In the Bayesian framework, one seeks to estimate the posterior probabilities for each state, based on prior probabilities calculated from the class labels. The class corresponding to the highest posterior is selected. An indirect representation of the posterior probability is the weight vector (W) which is calculated to train the classifier (given in equation (9)). In BLDA, the weight vector is computed recursively in contrast to the direct calculation used in LDA. In contrast, Perceptron Batch (PB) and nonlinear SVM aim to separate classes with the largest possible margin. A possible drawback of both techniques is the need to predefine a maximal number of iterations. The four algorithms were compared in terms of accuracy, error, bitrates, and computational complexity.

Classification error is defined as the ratio of erroneously emitted commands (N_e) to the total commands emitted (N_t) and is computed from the formula:

$$\text{Error} = \frac{N_e}{N_t} * 100\% \quad (1)$$

And accuracy, as such, is the ratio of correct emitted commands (N_c) to the total commands emitted is

$$\text{Accuracy} = \frac{N_c}{N_t} * 100\% = \left(1 - \frac{N_e}{N_t}\right) * 100\% \quad (2)$$

When assessing the performance of communication systems, Information Transfer Rates (ITRs) are given in terms of bit-rates. There are several bit-rate definitions in literature; among the first reported is that by Farwell and Donchin [14] defined as:

$$bit\ rate = V * R \tag{3}$$

where V is the classification speed in (symbols/minute) and R is the information carried by one symbol (bits/symbol), defined as:

$$R_{Farwell\&Donchin} = \log_2 N \tag{4}$$

where, N is the number of possible targets.

The second definition which is based on Shannon information theory for noisy channels was introduced in Wolpaw et al. [3] given as:

$$R_{Wolpaw} = \log_2 N + p * \log_2 p + (1 - p) \log_2 \frac{(1 - p)}{(N - 1)} \tag{5}$$

where, p is the probability of a target being correctly classified. In this study, the definition in equation (5) for computing the bit-rates was chosen because it takes accuracy into account, thus representing information transfer rates without assuming a faultless classifier; although at 100% accuracy, it is reduced to the definition in equation (4).

The principal objective of a P300 based algorithm is to detect target signals. In statistical terminology the algorithm estimates the probability of a certain data set containing a P300 wave. This study compares the previously mentioned set of algorithms in terms of the above mentioned parameters. As offline analysis is used to train and test the data, a parameter that indicates the computational complexity is required. To establish a comparison between the different methods, their runtimes were computed using Matlab profiler; the lower the runtime, the more feasible the method for online implementation on a small digital signal processing (DSP) board. Results were obtained using the 4-fold cross validation procedure. The setup implemented contained four sessions for each subject. As such, three labeled sessions were used to train the classifier and the fourth was used unlabeled to test it. At each run, a record of the number of correct emitted commands was kept to compute the average error rate and the bit-rate corresponding to each subject. A 4-fold Cross validation was used to obtain average values of errors and bit-rates (\pm) the standard deviation. The procedure was repeated for the different algorithms. Classification errors and bit rates were obtained for eight subjects. Four of the subjects (A, B, C, D) had neurological deficits, while the remainder (E, F, G, H) were healthy with no known neurological disorders. The data used in training and testing contained eight channels of EEG signals recorded from four midline electrodes (Cz, Pz, Fz, and Oz) and four parietal electrodes (P3, P4, P7, and P8). Each decision was emitted based on a probability comparison between six different datasets corresponding to the different visual stimuli that were elicited. At each correctly classified command, there are one true positive and five true negatives, and at each erroneously emitted command there are false positive and five false negatives.

The sensitivity and the specificity were calculated, respectively with the given below formula [15].

$$Sensitivity\ (SE) = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \tag{6}$$

$$Specificity\ (SP) = \frac{True\ Negatives\ (TN)}{True\ Negatives\ (TN) + False\ Positives\ (FP)} \tag{7}$$

Pattern Recognition Stages

An important step when presenting the proposed system is the implementation of the pattern recognition stages in the communication channel between the human brain and the computer.

Preprocessing

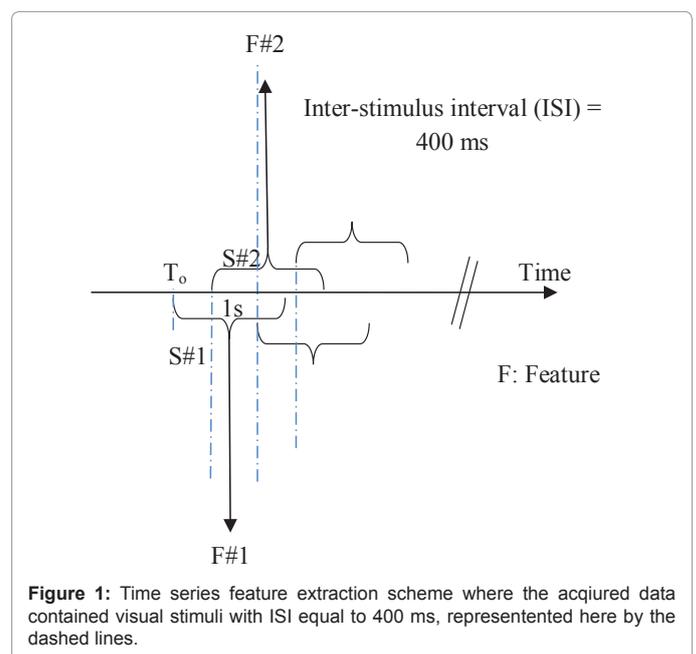
For BCI applications, the recorded data is preprocessed to reduce noise, artifacts, and dimension, prior to being fed to a machine learning algorithm. The data was preprocessed by six different preprocessing blocks. A high order notch filter was used to eliminate power line noise. A third-order Butterworth bandpass filter was used with lower and upper cutoff frequencies of 1 and 12 Hz, respectively. The cutoff frequencies were varied for each subject to identify the values that produced the best results. The data was then down sampled to 32 samples to reduce the dimension of the filtered data.

Stimuli were elicited every 400 ms, and due to the variable latency of the P300 component among subjects, extraction proceeded for one second after the onset of stimulus, which resulted in 600 ms of overlap as can be seen in Figure 1. Each trial was concentrated into a multidimensional array and sent to the next stage where it was multiplied by a window function to emphasize the late signal content. In the following stage, trials were scaled to a [-1,1] interval, and normalized to have a zero mean and a unity variance according to Equation (8).

$$z = \frac{x - \mu}{\sigma / \sqrt{n}} \tag{8}$$

where μ is the mean value, σ is the standard deviation, and n is the number of data points.

A whitening transform was applied to the extracted features. The data covariance matrix proportional to the identity was used for data whitening [5]. In order to obtain coordinate transformation, the mean (μ) was subtracted from the data as in equation (9).



$$\tilde{x} = \begin{bmatrix} \cdot \\ \cdot \\ x_i \\ \cdot \\ \cdot \end{bmatrix} - \mu \quad (9)$$

Then Eigenvalue-decomposition was applied to the data covariance matrix, as in equation (10):

$$[\lambda_i, v_i] = \text{eigen}(C) \quad (10)$$

where λ_i is the Eigen values matrix, v_i is the Eigenvectors matrix, and C is the data covariance matrix defined in Equation (11):

$$C = \frac{1}{N-1} \sum_{i=1}^N \begin{pmatrix} \tilde{x} \\ \tilde{x} \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{x} \end{pmatrix}^T \quad (11)$$

A new whitened data vector (X) is obtained by performing the transformation as shown in equation (12):

$$X = \frac{v D^{-\frac{1}{2}} \tilde{x}}{v^T} \quad (12)$$

where D is the diagonal matrix of the given values illustrated in (13):

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & \dots & 0 \\ 0 & \lambda_2 & \dots & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \dots & \lambda_n \end{bmatrix} \quad (13)$$

Machine Learning algorithms for classification: When evaluating uncertainties in data we often rely on probabilistic methods such as Bayes theorem [5].

The aim of any Bayesian algorithm is to approximate the probability of a state w given evidence x . Bayes theorem is defined as in equation (14).

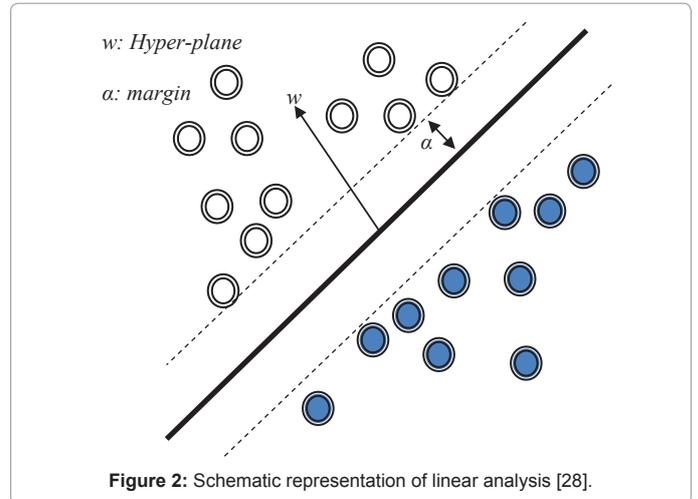
$$p(w|x) = \frac{p(x|w)p(w)}{p(x)} \quad (14)$$

where $p(w)$ is the probability of occurrence of state w , $p(x)$ is the probability of occurrence of event x , $p(w|x)$ is the probability of occurrence of state w given the event x , and $p(x|w)$ is the probability of occurrence of event x given the state w .

With the prior probabilities defined and the evidence approximated by a multivariate density function, the posterior probability is evaluated for each state and the class/command corresponding to the largest probability is selected according to equation (15).

$$\text{Posterior} = \frac{\text{Likelihood} \cdot \text{Prior}}{\text{Evidence}} \quad (15)$$

Classification is based on distinguishing one feature from another. If the features are well extracted and represent an event that occurred in the time series, the classifier is set to distinguish between two or more different states/classes and select the most probable event although this



process is sometimes contaminated by errors due to noise and artifacts in the EEG.

In this study, LDA, BLDA, and Perceptron programming were used for linear classification, while a nonlinear sigmoid SVM was implemented to train a nonlinear classifier. The LDA model assumed is simple, yet robust and sensitive to artifacts and noise. The objective is to evaluate a linear hyper-plane (w) with maximum margin (α) that separates two different classes (commands/states) (Figure 2).

When applying LDA, a sequence of observations given by the expression $\{x_i[n] = x_1 \dots x_N\}$, is extracted from the original signal, where N is the number of samples, and i is the number of input vectors. The true class labels vector of these observations is defined as $y_i \in \{-1, 1\}$. The linear classifier is given by:

$$F(x) = \text{sign}(w^T x + b) \quad (16)$$

Computation of the normal vector w that satisfies the maximum criterion function outlined in equation (17) gives the normal vector as shown in equation (18):

$$J(w) = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2} \quad (17)$$

$$w_i = S_i^{-1} (m_{i1} - m_{i2}) \quad (18)$$

where S is the scatter matrix defined as seen in equation (19):

$$S = \sum \begin{pmatrix} \tilde{x} \\ \tilde{x} \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{x} \end{pmatrix}^T \quad (19)$$

Performance measures: The performance evaluation of the BCI algorithm was measured in terms of classification error (or classification accuracy), and bit-rates were used to assess the information transfer characteristics. Published performance data are summarized in Table 1. To study the feasibility of BCI algorithms for online applications, the runtimes for each algorithm were computed using a 2.6 GHz Intel dual-core processor. The runtime of each algorithm was scaled to the maximum runtime of all methods at a specific data size.

Results and Discussion

The classification error and bit-rate against time for the disabled

ML algorithm	Neurophysiologic signal	Maximum bit-rate (bit/minute)	Accuracy (%)		Research group
			Average	Best	
SWDA	P300	Not reported	84	96	(Sellers and Donchin 2006)
Fuzzy logic	P300	Not reported	84	87	(Salimi-Khorshidi et al 2008)
BLDA	P300	64	~100	~100	(Hoffmann et al. 2008)
SWDA	P300	12.34	65.4	100	(Sellers et al. 2006)
AAPE*	Mu rhythm	Not reported	---	100	(Guger et al. 2001)

*Adaptive autoregressive parameter estimation

Table 1: Published performance data.

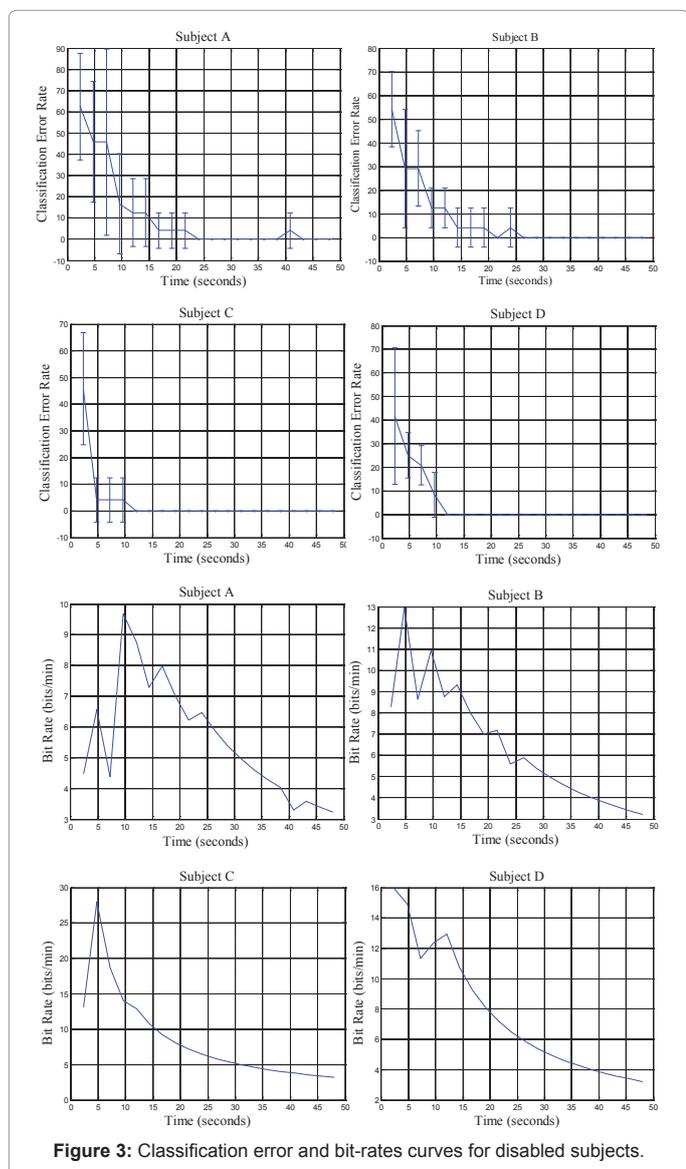


Figure 3: Classification error and bit-rates curves for disabled subjects.

and healthy subjects obtained with BLDA are shown in Figures 3 and 4. The remaining algorithms, LDA, PB, and SVM, came up with varying results but all converged to 100% accuracy except for subject E. Table 2 shows the confusion matrix obtained as a result of the decision made based on probability comparison. By looking at Figures 3 and 4 one

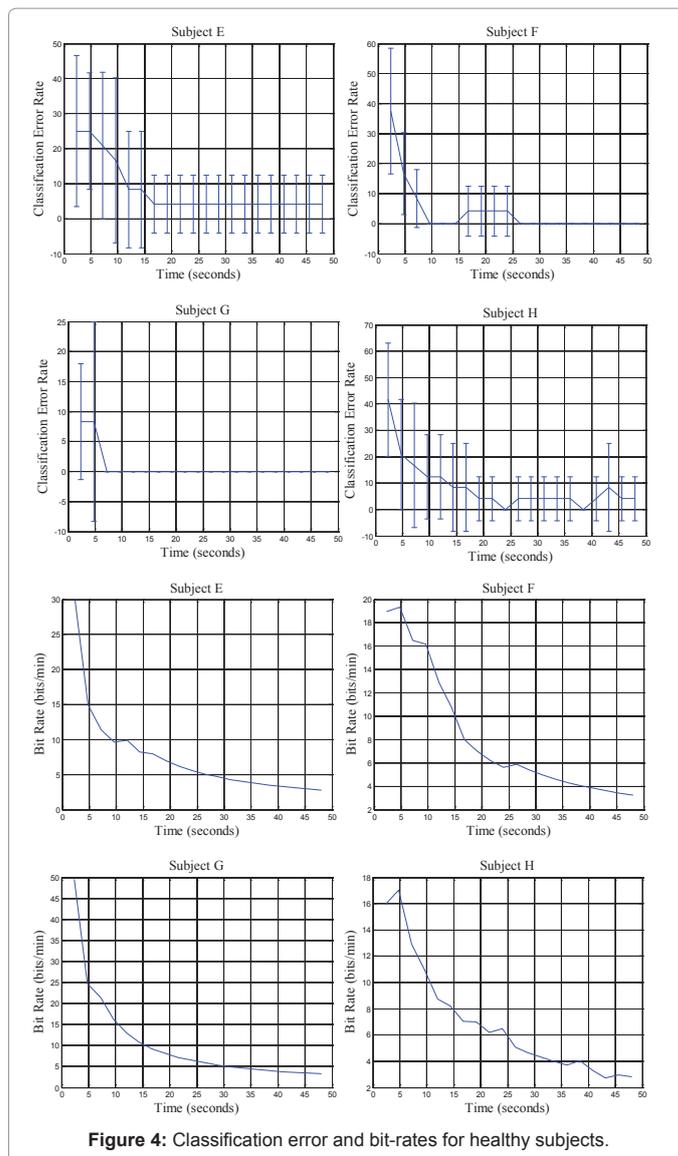


Figure 4: Classification error and bit-rates for healthy subjects.

	Positive	Negative
True	True positives (TP)=Number of correct emitted commands. BLDA: 903, LDA: 882, PB: 888, SVM: 863.	True Nagavite=5 • TP BLDA: 4515, LDA: 4410, PB: 4440, SVM: 4315.
False	FP= Number of erroneously emitted commands. BLDA: 57, LDA: 78, PB: 72, SVM: 97.	False Nagavite=5 • FP BLDA: 285, LDA: 390, PB: 360, SVM: 485.

Table 2: The confusion matrix of the four methods tested in this study.

can notice the inverse relation between the bit-rate and accuracy, this suggests that the more trials are incorporated in the decision, the higher the accuracy and the lower the bit-rate. But since 100% accuracy is approached it is assumed that the operating bit-rate for that accuracy is the one obtained at the same time interval. For example, subject A will operate a faultless classifier at 7.5 bits/minute and subject E at 12.5 bits/minute.

The maximum accuracy, the sensitivity, and the specificity, obtained for the different algorithms are presented in Table 3. It is fairly clear that, in terms of accuracy, all of the methods behave similarly. However,

Subjects	BLDA	LDA	PB	Nonlinear SVM
A, B, C, D, F, and G	100	100	100	100
E	95.8	95.8	95.8	95.8
H	100	100	95.8	95.8
SE (%)	76.01	69.41	71.15	64.02
SP (%)	98.75	98.26	98.40	97.80

Table 3: The maximum averaged classification accuracy (%) for the four algorithms and the corresponding sensitivity (SE) and specificity (SP) of each method.

Subject	BLDA		LDA		PB		Nonlinear SVM	
	MSE	bit-rate	MSE	bit-rate	MSE	bit-rate	MSE	bit-rate
A	1.12	10	2.70	8	1.61	7.5	3.43	6.5
B	0.59	13	1.26	8	0.76	11.2	0.95	13.2
C	0.08	28	0.11	28	0.19	19	0.34	14
D	0.22	16	0.66	17	0.31	17	0.62	15
E	0.66	30	1.12	16	0.69	19	1.08	19
F	0.15	19.3	0.15	19.3	0.27	19.3	0.27	19.3
G	0.006	50	0.02	50	0.08	26	0.12	22.3
H	0.72	17	0.91	15	1.46	20	3.83	7.5
Average for disabled subjects	0.502	16.75	1.182	15.25	0.717	13.675	1.335	12.175
Average for healthy subjects	0.384	29.075	0.55	25.075	0.625	21.075	1.325	17.025

Table 4: The averaged mean squared Error, and the maximum averaged bit-rate.

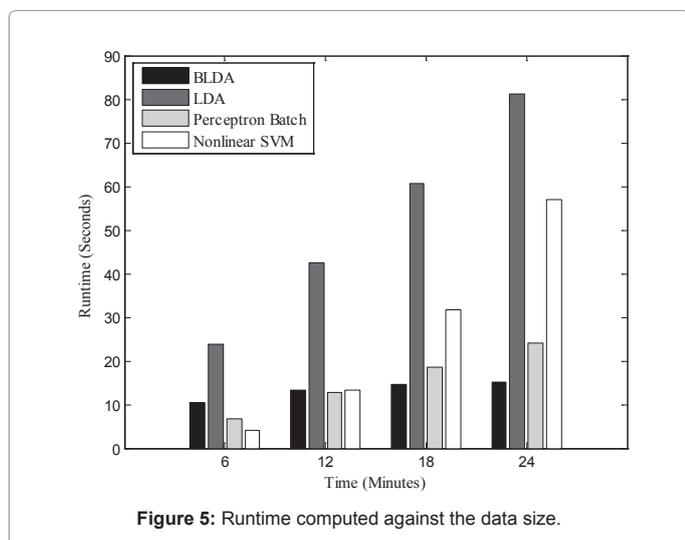


Figure 5: Runtime computed against the data size.

in terms of classification speed (bit-rate), variable results were obtained. The mean squared error (MSE) and the maximum bit-rate obtained for all subjects is presented in Table 4.

All of the results were averaged over the four sessions. The average bit-rates obtained with BLDA, LDA, PB, and nonlinear SVM for all subjects were 23 ± 13 , 20 ± 13.6 , 17.3 ± 5.6 , and 14.6 ± 5.5 bits/minute, respectively. This suggests that BLDA outperformed all the other methods in terms of both speed and accuracy since bit-rate and accuracy are in direct proportionality.

For offline analysis, all algorithms obtained fairly good results. However from a developer's point of view, a system is best when trained online and operated on a small portable hardware. Thus noncomplex and fast algorithms need to be developed. To access the feasibility of

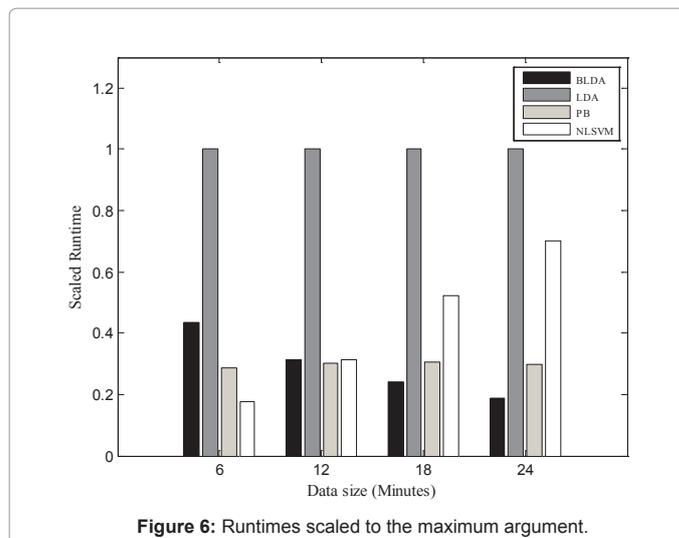


Figure 6: Runtimes scaled to the maximum argument.

Number of Channels	Bit-rate without data whitening			Bit-Rate with data whitening		
	4	8	32	4	8	32
Subject A	8	8	13	8	10	9
B	7	10	11	6	13	9
C	22	25	22	22	28	20
D	15	19	30	16	16	17
E	26	25	34	26	30	13
F	22	22	39	19	19	26
G	38	50	64	39	50	43
H	17	19	17	19	17	15

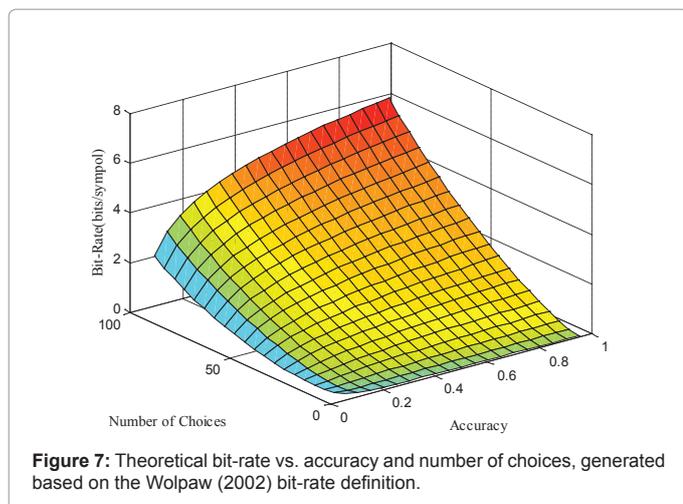
Table 5: The effect of data whitening on classification speed.

each method used for such a task, the runtime was computed against the data size as shown in Figure 5.

It is obvious that as the data size increases, BLDA and Perceptron batch converge faster than the other two methods. The study employed 18 minutes of data for training and 6 minutes for testing. For this set, BLDA had the fastest runtime (14.5 seconds) and LDA had the slowest (60 seconds); this can be explained by the fact that LDA computes the inverse matrix directly to obtain the weight vector while in BLDA hyper-parameters are computed recursively to obtain the weight vector. Runtimes were scaled to the maximum argument of all methods to demonstrate how the runtimes vary as the data size increases (Figure 6).

The runtimes in Figure 5 were averaged across four runs for each algorithm. From Figures 5 and 6, it can be seen that BLDA and PB converge faster and are less affected by the data size in comparison with the other two algorithms.

The four algorithms tested varied in runtimes as the data set size increased (Figures 5 and 6). However, BLDA and PB were not significantly affected by the sample size and exhibited robustness in the training phase. For example, with BLDA, when the data size increased from 6 to 12 minutes, the average runtime increased by only 3 seconds. On the other hand, using LDA the runtime increased almost 20 seconds. The average runtime for BLDA was 13 ± 2 seconds, and the average runtime of PB was 15.6 ± 6 seconds. For LDA and Nonlinear SVM the runtimes were in the range of 1-2 minutes for the 24 minutes of data set size.



Data size (minutes)	BLDA	LDA	PB	Nonlinear SVM
6	11	24	7	5
12	13	43	12	13
18	14	63	17	31
24	14	81	24	66

Table 6: Runtimes (Seconds) for different methods.

The classification speed (bit-rate) seemed to increase when data whitening was used in preprocessing. This was only valid for data with a low dimension (eight and four channels). With higher dimension data, whitening seemed to decrease the classification speed. In a BCI setup, it is more convenient to use a low number of channels. Table 5 shows the effect of data whitening on classification speed.

Several studies reported that the amplitude of the P300 increases proportionally with the number of choices [16,23]. On the other hand, as the number of choices is increased, it is fairly obvious that the probability of error increases. Nijboer et al. [24] reported a P300 speller system using 6x6 and 7x7 matrices (choices). The study included offline and online classifications, and reported higher accuracies for offline analysis. Theoretically, speed (bit-rate) increases with increased number of choices (Figure 7). However, there is a practical limitation imposed by the fact that a high number of choices require higher dimensional features for training and testing.

The performance of a BCI system, as expected, depends on the machine learning algorithm it uses. Many algorithms have been developed and utilized in P300 detection. The question of which is better is never simple due to the performance variability observed among subjects. Generally the method that requires minimal training data and needs less user intervention is better. Moreover a better method takes less time to converge. Due to the rapid development of fast computers, multiple algorithms can be used in parallel. The methods used here all obtained high performance in terms of accuracy. However two of them might be of significant importance in future BCI research: BLDA and PB, due to their low runtimes which would make them practical for real-time applications has been mentioned in Table 6. Other methods include Hidden Markov Models (HMMs), ICA, and Wavelets Packets Transform (WPT). Obermaier et al reported the use of HMMs for online classification of motor imageries [25]. Hung et al. [26] used ICA

in pre-classification and reported an increase in accuracy. Limitations on these methods include the necessity for knowing the number of original sources for ICA, and choosing an appropriate Wavelet type and the number of scales for WPT. These methods were not implemented in the presented P300-based BCI and need further research and testing.

Conclusions

People suffering from neuromuscular dysfunction may use a P300-based BCI to communicate with their environment quite successfully. This study showed that people suffering from neuromuscular disorders will perform slightly lower than healthy subjects (Table 4), future development of this work needs to include a larger pool of subjects to validate this claim and to test whether the difference in performance is consistent between healthy and disabled subjects. All algorithms adopted in this study produced acceptable levels of performance but two of the four algorithms (BLDA, and PB) were superior in terms of minimal runtimes as their runtimes were found to be much lower than the actual data length when acquiring the time vector online. As a result, implementing both BLDA and PB would provide the best choice. In general, all methods performed accurately but slowly. The reliance on the P300 wave in brain computer communication defines the information transfer characteristics since P300 is related to time in latency and duration and is dependent on an external stimulus. Improvement of information transfer rates can be done either by increasing the number of choices (Figure 7), which is practically limited, or by increasing the number of commands that can be emitted in one minute. Further research on strategies is needed to develop high-speed online algorithms for enhanced user convenience.

Acknowledgement

Many thanks go to the Swiss Federal Institute of Technology for making their data and tools available online.

References

1. Teplan M (2002) Fundamentals of EEG Measurement. Measurement Science Review 2.
2. Krepki R, Curio G, Blankertz B, Muller KR (2007) Brain-Computer Interface—The HCI communication channel for discovery. Int J Hum-Comput St 65: 460-477.
3. Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain-Computer Interfaces for communication and control Clin Neurophysiol 113: 767-791.
4. Cipresso P, Carelli L, Solca F, Meazzi D, Meriggi P, et al. (2012) The use of P300-based in amyotrophic lateral sclerosis: from augmentative and alternative communication to cognitive assessment. Brain and Behavior 2: 479-498.
5. Duda RO, Hart PE, Stork DD (2000) Pattern Classification. Wiley Interscience.
6. Kenji N, Yasuki K, Akihiro H (2007) A Brian Computer interface Based on FFT and Multilayer Neural Network- Feature Extraction and Generalization. International Symposium on Intelligent Signal processing and Communication Systems 826-829.
7. Navin Lal T, Schröder M, Hinterberger T, Weston J, Bogdan M, et al. (2004) Support Vector Channel Selection in BCI. IEEE 51: 1003-1010.
8. Townsend G, LaPallo BK, Boulay CB, Krusienski DJ, Frye GE, et al. (2010) A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns. Clin Neurophysiol 121: 1109-1120.
9. Leow RS, Ibrahim F, Moghavvemi M (2007) Development of a Steady State Visual Evoked Potential (SSVEP) - Based Brain Computer Interface (BCI) system. International Conference on Intelligent and Advanced Systems 321-324.
10. Reza Fazel-Rezai, Brendan ZA, Christoph Guger, Sellers EW, Sonja CK, et al. (2012) P300 brain computer interface: curren challenges and emerging trends. Frontiers in Neuroengineering.

11. Pieter-Jan Kindermans, David Verstraeten, Benjamin Schrauwen (2012) A Bayesian Model for Exploiting Application Constraints to Enable Unsupervised Training of a P300-based BCI. *PLoS ONE* 7: e33758.
12. Guger C, Scogl A, Neuper C, Walterspacher D, Strein T, et al. (2001) Rapid Prototyping of an EEG-Based Brain-Computer Interface (BCI). *IEEE Trans Neural Syst Rehabil Eng* 9: 49-58.
13. Hild KE, Orhan U, Erdogmus D, Roark B, Oken B, et al. (2011) An ERP-based Brain-Computer Interface for text entry using Rapid Serial Visual Presentation and Language Modeling, *Proceedings of the ACL-HLT 2011 System Demonstrations* 38–43, Portland, Oregon, USA, 21 June 2011.
14. Farwell LA, Donchin E (1988) Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr Clin Neurophysiol* 70: 510-523.
15. MacKay DJC (1992) Bayesian Interpolation. *Neural Computation* 4: 415-447.
16. Hoffmann U, Vesin JM, Ebrahimi T, Diserens K (2008) An Efficient P300-based Brain-Computer Interface for disabled subjects. *J Neurosci Methods* 167: 115-125.
17. Ye J, Li Q (2005) A Two-Stage Linear Discriminant Analysis via QR-Decomposition. *IEEE Xplore* 27: 929-941.
18. Erfanian A, Erfani A (2004) ICA-based classification scheme for EEG-based brain-computer interface: the role of mental practice and concentration skills. *Conf Proc IEEE Eng Med Biol Soc* 1: 235-238.
19. Park S, Lee H, Choi S (2003) ICA+OPCA For Artifact-Robust Classification of EEG Data. *Neural Networks for Signal Processing* 585-594.
20. Qin J, Li Y, Cichocki A (2005) ICA and Committee Machine-Based Algorithm for Cursor Control in a BCI System. *Lecture Notes in Computer Science* 3496: 973-978.
21. Sellers EW, Donchin E (2006) A P300-based brain-computer interface : Initial tests by ALS patients. *Clin Neurophysiol* 117: 538-548.
22. Blumberg J, Rickert J, Waldert S, Schulze-Bonhage A, Aertsen A, et al. (2007) Adaptive Classification for Brain Computer Interfaces. *Conf Proc IEEE Eng Med Biol Soc* 2007: 2536-2539.
23. Sellers EW, Krusienski DJ, McFarland DJ, Vaughan TM, Wolpaw JR (2006) A P300 Event-Related Potential Brain-Computer Interface (BCI): The effect of matrix size and Inter stimulus interval on performance *Biol Psychol* 73: 242-252.
24. Nijboer F, Sellers EW, Mellinger J, Jordan MA, Matuz T, et al. (2008) A P300-based brain-computer interface for people with amyotrophic lateral sclerosis. *Clin Neurophysiol* 119: 1909-1916.
25. Obermaier B, Guger C, Neuper C, Pfurtscheller G (2001) Hidden Markov models for online classification of single trial EEG data. *Pattern Recogn Lett* 22: 1299-1309.
26. Hung CI, Lee PL, Wu YT, Chen LF, Yeh TC, et al. (2005) Recognition of motor Imagery Electroencephalography Using Independent Component Analysis and Machine Classifiers. *Ann Biomed Eng* 33: 1053-1070.
27. Salimi-Khorshidi G, Nasrabadi A, Golpayegani M (2008) Fusion of classical P300 detection methods' inferences in a framework of fuzzy labels. *Artif Intell Med* 44: 247-259.
28. Müller KR, Anderson CW, Birch GE (2003) Linear and Nonlinear Methods for Brain-Computer Interfaces. *IEEE Xplore* 11: 165-169.