THESIS

COMPARISON OF EEG PREPROCESSING METHODS TO IMPROVE THE PERFORMANCE OF THE P300 SPELLER

Submitted by
Zachary Cashero
Department of Computer Science

In partial fulfillment of the requirements
For the Degree of Master of Science
Colorado State University
Fort Collins, Colorado
Summer 2011

Master's Committee:

Advisor: Charles Anderson
Co-advisor: Thomas Chen

Stuart Tobet
Asa Ben-Hur

ABSTRACT

COMPARISON OF EEG PREPROCESSING METHODS TO IMPROVE THE PERFORMANCE OF THE P300 SPELLER

The classification of P300 trials in electroencephalograhic (EEG) data is made difficult due the low signal-to-noise ratio (SNR) of the P300 response. To overcome the low SNR of individual trials, it is common practice to average together many consecutive trials, which effectively diminishes the random noise. Unfortunately, when more repeated trials are required for applications such as the P300 speller, the communication rate is greatly reduced. Since the noise results from background brain activity and is inherent to the EEG recording methods, signal analysis techniques like blind source separation (BSS) have the potential to isolate the true source signal from the noise when using multi-channel recordings. This thesis provides a comparison of three BSS algorithms: independent component analysis (ICA), maximum noise fraction (MNF), and principal component analysis (PCA). In addition to this, the effects of adding temporal information to the original data, thereby creating time-delay embedded data, will be analyzed. The BSS methods can utilize this time-delay embedded data to find more complex spatio-temporal filters rather than the purely spatial filters found using the original data. One problem that is intrinsically tied to the application of BSS methods is the selection of the most relevant source components that are returned from each BSS algorithm. In this work, the following feature selection algorithms are adapted to be used for component selection: forward selection, ANOVA-based ranking, Relief, and recursive feature elimination (RFE). The performance metric used for all comparisons is the classification accuracy of P300 trials using a support vector machine (SVM) with a Gaussian kernel. The results show that although both BSS and feature selection algorithms can each cause significant performance gains, there is no added benefit from using both together. Feature selection is most beneficial when applied to a large number of electrodes, and BSS is most beneficial when applied to a smaller set of electrodes. Also, the results show that time-delay embedding is not beneficial for P300 classification.

K-12 SUMMARY

Since I was involved with the GK-12 program, one specific objective is the ability to communicate graduate level research at the K-12 level. This summary is intended to provide an overview of my thesis work targeted at K-12 students. There is a condition called "locked-in syndrome" that occurs when a person has become completely paralyzed and can no longer communicate. This could result from an accident or from a disease that causes paralysis. When a person is completely paralyzed and does not even have control of their facial muscles, there is no way to communicate even a simple "yes" or "no" answer. They end up becoming trapped in their own body. A brain computer interface (BCI) has the ability to restore basic communication to a person like this through their brain waves. Electrical activity is constantly being generated by neurons in the brain. The BCI system observes this brain activity using sensors that are put into a cap that the person wears on their head. The person can change their brain waves depending on what they are thinking, and the BCI system can pick up on these changes to perform some action for the user, such as driving a wheelchair, moving a prosthetic arm, or controlling their computer. Since it is only using the person's thoughts and does not require any muscle movements, a paralyzed person can successfully use a BCI. The basic communication and control that a BCI can provide will greatly enhance the quality of life for a locked-in patient. The work in this thesis focuses on one specific BCI called the P300 speller. This spelling program allows the person to type one letter at a time by flashing each letter on the screen in front of them. If the person is looking for the specific letter that they want to type, each time that letter is flashed, a spike in the brain activity is generated. By detecting this spike, it is possible to determine which letter the person wants to type. However, it is challenging to detect this spike because it is very small, and it is mixed in with a lot of other brain activity. The other brain activity is referred to as "noise" because it makes it difficult to see the spike, just as a lot of extra noise can make it difficult to hear someone else speak. The brain activity is recorded with many sensors that are placed around the head. The work in this thesis compares different mathematical approaches for combining the signals from all of these sensors in order to reduce the noise. If the noise can be reduced, the spike is easier to detect, and this results in a BCI with a better performance that is

more accurate when choosing the letter that the person wants to type. The results show that using certain mathematical approaches, a person can type up to 50% faster.

TABLE OF CONTENTS

Chapter 1

Introduction

The field of brain-computer interfaces (BCI) has emerged from the desire for new assistive technology, targeted at patients who are paralyzed and have lost all means of communication. Some specific target populations are patients with spinal cord injury or patients with amyotrophic lateral sclerosis (ALS), which is a neurodegenerative disease that can result in the loss of voluntary muscle movement. Many patients may still retain some voluntary control of their facial muscles which can be used as a reliable trigger for communication or the control of some external device. However, in some patients, the disease can progress to a point that will cause "locked-in" syndrome, which is a condition where the patient is awake and fully aware but cannot communicate with the outside world due to complete paralysis. In these cases, a BCI has the potential to establish a communication channel directly from the patient's brain signals to the computer. There are two factors that must be considered for a BCI system: finding a brain signal that the patient can reliably and voluntarily control without the use of any muscular movements, and developing the analysis software that must detect these specific brain signals. Although BCIs currently only allow for limited communication, they have shown good potential for the basic control of a mouse cursor, a speller program, a wheelchair, a prosthetic arm, or some other external device, which can greatly increase the quality of life for a locked-in patient.

The brain signals used for a BCI can be obtained either through invasive or non-invasive recording methods. Research has been conducted using implanted electrodes on primates [1, 2, 3], and later, the BrainGate$^{\mathrm{TM}}$Neural Interface System was developed by Cyberkinetics Neurotechnology Systems, Inc. and successfully implanted in humans [4]. These implanted electrodes allow for a very accurate reading of the electrical activity over a small portion of the brain. However, non-invasive methods are attractive since they do not require any surgical procedures, and the performance has been shown to be comparable to implanted electrodes by Wolpaw and McFarland [5] when using more a sophisticated adaptive algorithm. In this work, the data is obtained through non-invasive electroencephalographic (EEG) recordings from a set of electrodes placed directly on the scalp using the International 10-20 system as shown in Figure 1.1. Each electrode detects the electric potential of synchronized neuronal activity occurring in that area of the brain. Since the electric potentials must pass through the skull, the EEG signals are inherently very noisy, which presents many challenges for signal analysis and pattern recognition.

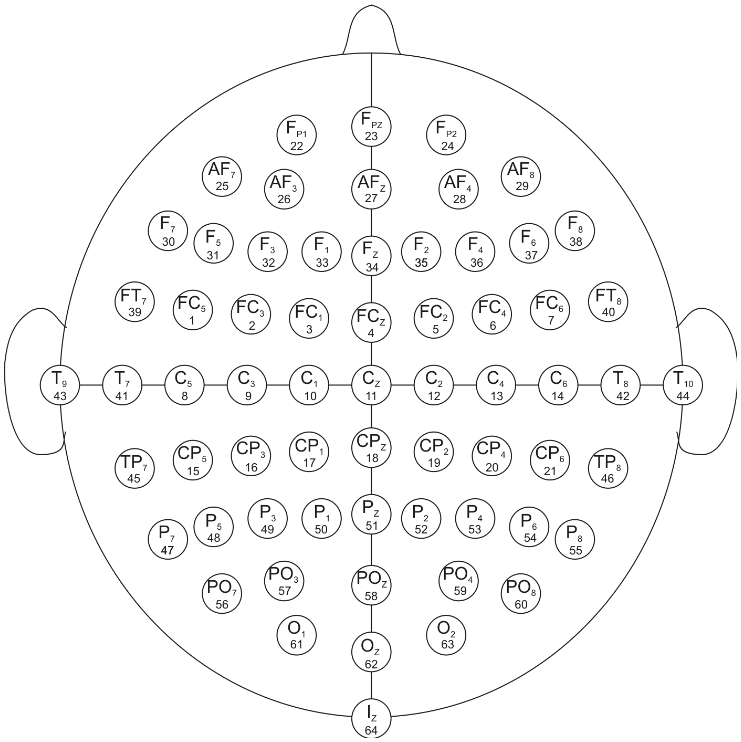


Figure 1.1: This shows the electrode placement for a 64-channel EEG system using the International 10-20 system. The image is taken from [6].

There are many different approaches for BCIs, each having their own advantages and disadvantages, but they can generally be grouped into two categories. In the first category, the user voluntarily switches between a small set of mental tasks that each produce different brain patterns, where each mental task is associated with a specific action. The most commonly used tasks are motor imagery, such as imagining a hand or foot moving. The second category, which this thesis targets, uses evoked responses from external stimuli that are presented to the user. The user focuses their attention on a specific stimulus that elicits an event-related potential (ERP) in the EEG signals. The computer detects this ERP to determine the user's desired action.

1.1 P300 Speller

One of the more well studied ERPs is the P300 response, which is characterized by a large positive deflection in the voltage starting about 300 ms after the onset of the stimulus, as shown in Figure 1.2. The P300 response is elicited by the "oddball" paradigm, in which repeated stimuli are presented to the user, and there is a specific target stimulus that rarely occurs among the more common non-target stimuli. Each time the target stimulus is presented to the user, the P300 response appears in the EEG signals. Although the P300 response is known to occur with different forms of stimuli, such as auditory stimuli [7], only visual stimuli is considered here.

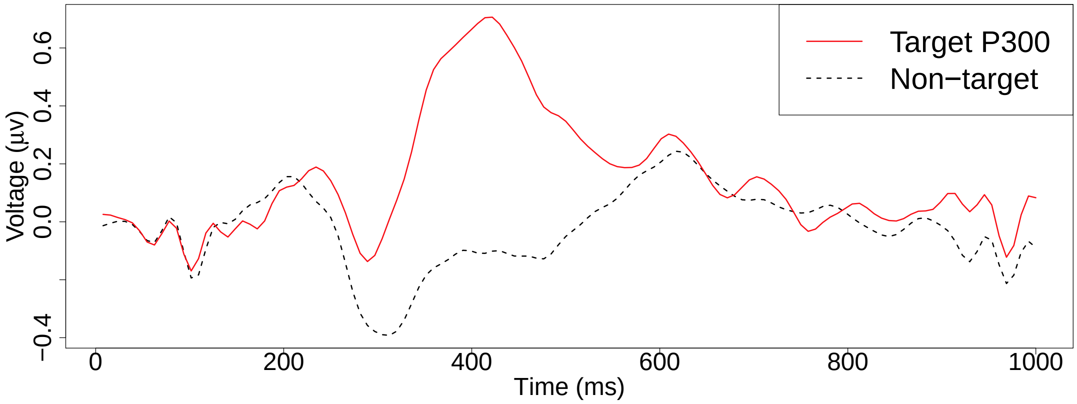
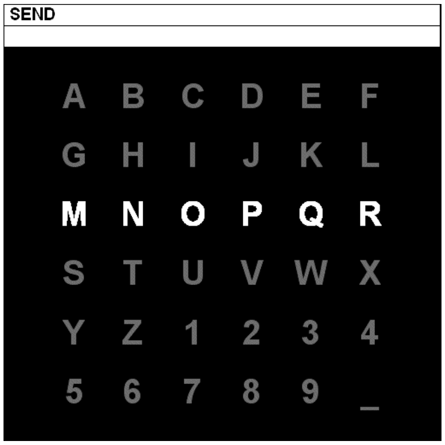


Figure 1.2: This plot shows the averaged P300 response at electrode Cz for Subject C, displaying a large positive peak from about 300–600 ms. This data was recorded at the Colorado State University Occupational Therapy lab using a 32-channel Biosemi system. More detail is given in Section 3.1.

Farwell and Donchin [8] first utilized the P300 response to create a BCI that allows the user to type a single letter at a time, referred to as the P300 speller. A 6 × 6 grid of characters (although it can be larger) is displayed to the user, as shown in Figure 1.3. The rows and columns of this grid flash at a constant rate, about eight times per second. The user must focus their attention on the letter that they want to type. A P300 response is elicited each time the row or column containing the target letter is flashed. This falls into the oddball paradigm because the random flashing is unpredictable but expected, and the probability of the target stimulus is only $\frac{1}{6}$ when using a 6 × 6 grid. It is then possible to identify the target letter from the intersection of both the target row and target column. Therefore, character recognition is broken up into two distinct tasks: the classification of the target row, and the classification of the target column. To classify a row, a window of data (e.g. a one-second window) is taken directly after the flashing of each of the six rows. The P300 response should be visible in only the row containing the target letter. Therefore, the row associated with the window of data that most closely resembles a P300 is determined to be the target row. The same concept is used to find the target column. The target row and column determine the desired letter, and this is referred to as character classification.



SEND

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

Figure 1.3: This shows a 6 × 6 P300 speller grid that was used for the BCI Competition III. The image is taken from [9].

One of the main challenges of P300 classification is the low signal-to-noise ratio (SNR). The P300 response is hard to distinguish amongst the background noise, caused by ongoing electrical activity in the brain. Therefore, the low SNR is inherent to this problem and is usually overcome by averaging together many subsequent trials. Figure 1.4 shows 20 individual target P300 trials, and it is clear to see how much noise is included in single trials and how difficult it is to find the true P300 signal. Figure 1.5 shows the effect of averaging together both target and non-target trials. If the background noise is assumed to be random Gaussian noise, as more trials are averaged together, the noise is effectively diminished, which makes the true P300 response more prominent. Although this makes the classification task easier, the drawback is that the system needs many more repeated trials to make one decision, resulting in a speed-accuracy tradeoff. In the context of the P300 speller, the user must stare at their target letter until the system has collected enough repeated trials to make the decision. This reduces the communication rate, and therefore, one of the current goals of this field is to reliably detect P300 responses using fewer averaged trials, with the eventual goal of single-trial classification.

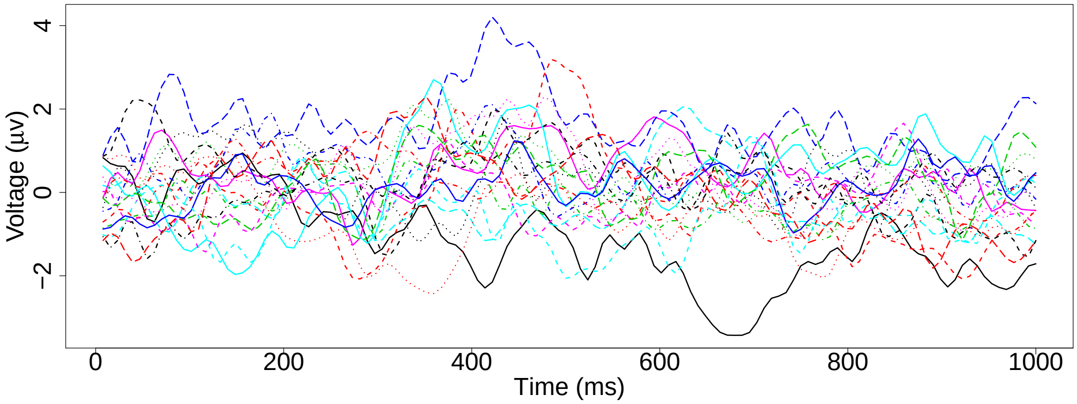
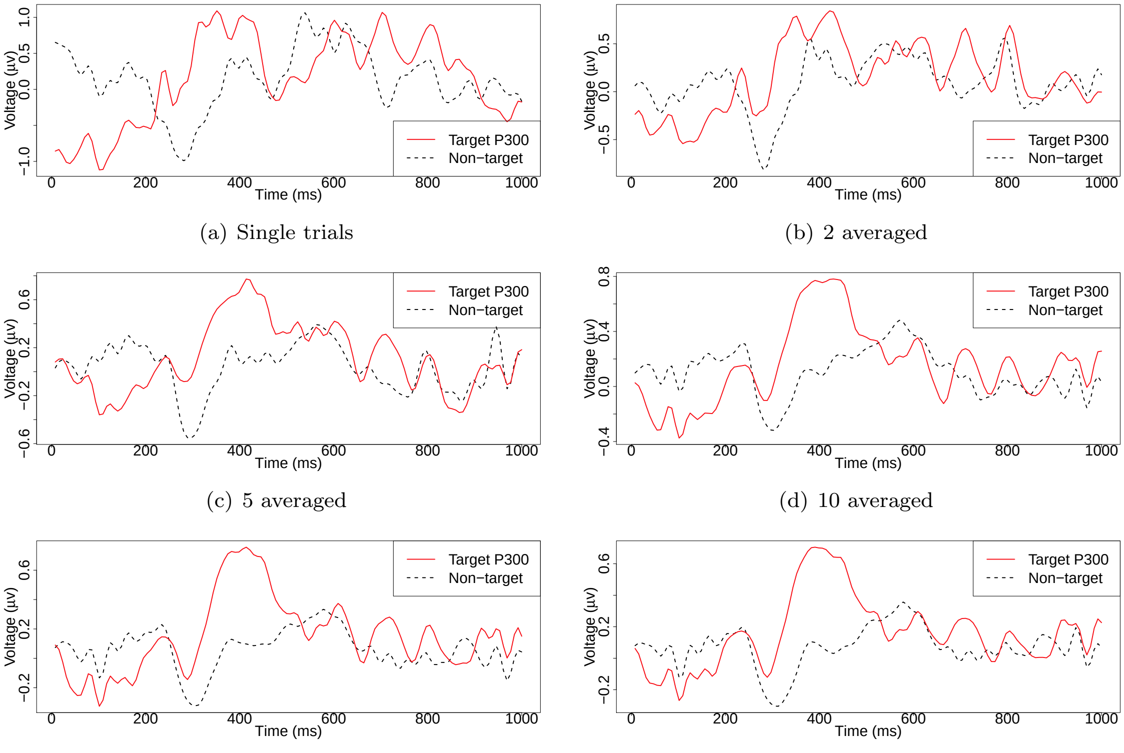


Figure 1.4: This plot shows 20 randomly selected individual P300 trials recorded from Subject C plotted on top of each other. It is clear to see the amount of noise in a single trial, which makes it difficult to detect the P300 signal.



(a) Single trials

(b) 2 averaged

(c) 5 averaged

(d) 10 averaged

(e) 15 averaged

(f) 20 averaged

Figure 1.5: This plot shows the effects of averaging trials together. The same 20 target trials from Figure 1.4 are averaged together and shown with 20 non-target trials. As more trials are averaged together from (a) - (f), the distinction between target and non-target trials is made more clear.

1.2 Related Work

1.2.1 State of the Art

A literature review of the field does not show any single P300 detection system to be the state of the art. Krusienski, et al., [9] report the results of a comparison of different classifier algorithms, which shows that stepwise linear discriminant analysis (SWLDA), a fairly simple linear classifier, and support vector machines (SVMs) perform well compared to the other classifiers. Unfortunately, it is difficult to directly compare between many of the approaches presented in the literature because they use data recorded from different labs and different subjects, which can vary greatly. There are two well known benchmark P300 datasets provided from the BCI Competition II [10] and the BCI Competition III [11]. Several different approaches were able to achieve 100% accuracy using only 4-8 averaged trials [12, 13, 14, 15] on the BCI Competition II data. Kaper, et al., [12] used a straightforward approach with a Gaussian SVM to achieve perfect results with 5 averaged trials. Xu, et al., [13] applied independent component analysis (ICA) using *a priori* information to select the best components to enhance the target responses. They then used a fairly simple peak-picking method to obtain 100% classification with 5–8 averaged trials. Bostanov [14] proposed a distinctly different approach that used a continuous wavelet transform to extract features from each trial and classified with linear discriminant analysis (LDA) to achieve perfect results with 6 averaged trials. The dataset from the BCI Competition III proved to be more challenging, with the best results coming from Rakotomamonjy and Guigue [16], who employed an ensemble of SVMs to obtain an accuracy of 96.5% using 15 averaged trials. The authors use an ensemble of 17 different SVM classifiers each trained on a different portion of the training instances in order to be more robust to variability between trials. This short list shows that there are many different approaches that all perform well, and it is not immediately clear whether one approach is better than the rest across multiple subjects.

Of course, the real test is to look at the online performance with actual patients. Some initial studies have been performed on ALS patients [17, 18] that show the P300 speller is a plausible system to use with severely disabled patients. Using three ALS patients, Sellers and Donchin [17] use a simplified P300 BCI to choose between only four options. The results show that two of the ALS patients were able to achieve offline accuracies comparable to the control group of able-bodied patients. Nijboer, et al., [18] found that four ALS patients with severe paralysis were able to achieve a mean online accuracy of 79% using the P300 speller interface. They also showed that the P300 response remained stable over a period of 40 weeks in all patients. These studies are promising and indicate that the P300 speller is a viable option for text communication with severely disabled patients.

Although the P300 speller has been studied extensively and is one of the more well established BCI systems, a recent review of the field by Mak, et al., [19] concludes that more work still needs to be done to optimize the speed, accuracy, and consistency before the P300 speller is practical to use with disabled patients. This becomes even more relevant when considering that ALS patients can display widely varying ERP responses between subjects. Paulus, et al., [20] found that 12 out of 16 ALS patients displayed abnormal P300 responses in regards to the latency, shape, and amplitude. This suggests that a reliable BCI system must be able to adapt to the unique responses of each subject's ERP and be robust enough to handle the variations between trials within a subject. It is standard practice to train the BCI system for each new subject, allowing it to only learn the characteristics of that individual's ERP. Therefore, some approaches might have difficulty if they use *a priori* information to make assumptions about the temporal and spatial characteristics of the standard P300 response, especially when applied to abnormal ERPs from ALS patients.

1.2.2 Blind Source Separation

Blind source separation (BSS) methods are based on the assumption that the observed signals from a multi-channel recording are produced from a mixture of several distinct source signals. In the context of EEG recordings, many spatially distinct brain sources are believed to contribute to the overall observed EEG signal. ICA, PCA, and MNF have all shown to be successful at removing artifacts in EEG [21, 22, 23] through BSS. Artifacts are considered to be any sort of EEG contamination resulting from biologically generated or external sources. Some examples are muscle activity, eye blinks, or 60 Hz line noise. Each of these BSS methods are known to isolate the artifact activity into individual source components. Compared to P300 source extraction, artifact extraction is a much easier task since the contaminating signals are usually large relative to the ongoing EEG activity. When used for P300 classification, only ICA has been applied and shown to isolate the P300 signal into source components [13, 24, 25, 26, 27]. The literature lacks comparative studies of ICA to other simpler methods like PCA and MNF.

Makeig, et al., [28, 29, 30] applied ICA to averaged ERP trials and confirmed that the observed signal is comprised of many spatially distinct and independent brain processes. Therefore, the application of BSS to P300 trials has the potential advantage of finding the corresponding spatial filters to isolate these independent sources. Xu, et al., [13] applied ICA to P300 data and used *a priori* information about the spatial and temporal characteristics of the standard P300 response to select the most relevant source components. Their algorithm selects components with larger amplitudes from 250–400 ms and stronger contributions from electrodes Cz, C1, and C2 that were later used for classification. Although it worked well for that single subject, as explained above, the disadvantage of using *a priori* information is that each subject's P300 response can vary significantly, especially in ALS subjects. Piccione, et al., [25] and Li, et al., [27] also applied ICA and both used different forms of *a posteriori* template matching to select the most relevant source components. Wang and James [26] applied spatially constrained ICA to P300 trials also using an *a posteriori* template matching algorithm, but it requires the manual selection of the best component to use as a template. Their results showed that this method increased character classification from 51.6% to 96.8% on the BCI Competition II dataset. Hill, et al., [24] classified auditory P300 trials with and without ICA to find that ICA generally improved the accuracies across all subjects, by up to 14% over the original data without applying ICA. They also applied recursive feature elimination (RFE) [31] to the resulting source components but found that although many components did not contribute to the performance, the elimination of these irrelevant components did not increase the accuracy.

Since traditional ICA (referred to here as spatial ICA) must be used on multi-channel recordings, it is not possible to use with only a single channel. However, Davies and James [32] introduced the concept of single channel ICA that utilizes time-delay embedded data. Time-delay embedded data uses the original data from one channel lagged by one or more time samples to introduce a new lagged dimension, essentially creating a new 'channel' that is utilized by the BSS method. When using only a single channel with lagged data, ICA produces components which are a linear mixture of the lagged time samples without any spatial information, thereby creating a purely temporal filter. The authors applied single channel ICA to ictal EEG recordings [32] and P300 recordings [33] to find that it is able to successfully separate meaningful sources. However, quantitative analysis showed that it is inferior to spatial ICA, which is to be expected since more data is available in a multi-channel recording. James, et al., [34] further extended this idea to a create spatio-temporal ICA method that uses time-delay embedded data from multi-channel EEG recordings to obtain a set of source components derived from spatio-temporal filters. Their results show that spatio-temporal ICA better isolates the ictal activity than traditional spatial ICA. Davies, et al., [35] use the same spatio-temporal ICA algorithm applied to P300 data, and they visually show that it is able to extract a source component resembling P300 activity. However, no quantitative analysis is performed.

1.3 Contributions

This work contributes to the field of P300 classification by analyzing several preprocessing techniques as a way of maximizing the meaningful information from the original data. In the following experiments, the classifier is kept constant to best isolate the effects of the different preprocessing steps. Specifically this work looks at the contribution of several BSS methods, ICA, PCA, and MNF. It is hypothesized that the BSS methods will isolate the meaningful P300 information into several source components, necessitating the selection of only the most relevant components. The component selection process is performed by the application of different feature selection algorithms adapted for this task: forward selection, ANOVA ranking, Relief [36], and SVM recursive feature elimination [31]. Using these algorithms, no *a priori* information is used and no assumptions are made about the characteristics of the P300 response. The only criteria is to find the most distinguishing information between target and non-target trials. The feature selection algorithms are also applied to the original data without BSS for channel selection in order to better understand the contribution from these algorithms. The following experiments are run on EEG data using the full set of electrodes and on EEG data using only a subset of 8 electrodes that have already been found to be meaningful for P300 classification [37]. Although a full set of electrodes provides more information, a smaller set of electrodes is desirable for practical reasons due to the lower cost and higher usability for in-home use with patients. The preprocessing steps exhibit different performance trends depending on the number of available electrodes, and the differences are explored in these experiments.

Another contribution of this work is to study the effects of time-delay embedding by adding lags to the data. The inclusion of temporal information allows the BSS methods to find more complex spatio-temporal filters, rather than purely spatial filters. Spatio-temporal filters have the potential to account for phase delayed coherence between electrodes. If activity in two separate areas of the brain are synchronized with a phase delay, the spatial BSS methods cannot recognize this. An ERP analysis by Makeig, et al., [38] using ICA found several source components containing essentially the same information except with a phase delay. The spatial BSS methods isolate these as independent components even though they are clearly dependent on each other. Therefore, the effects of temporal information in addition to spatial information are explored in the context of P300 classification.

The main objective of this work is to find a set of automated preprocessing steps that can be applied to the raw data to best extract meaningful ERP components from noisy EEG data. Utilizing time-delay embedded data, BSS, and feature selection algorithms, the effects of each of these steps will be analyzed to gain insights into the preprocessing steps that are most beneficial across different subjects, ensuring that the methods are able to adapt to inter-subject differences by making no assumptions about the characteristics of the P300 response. Although the experiments here use a support vector machine (SVM), the preprocessing steps are independent of the classification algorithm and can potentially be used in other systems, but it remains to be seen how well the results generalize to other classifiers. It is important to remember that the other components of the BCI system require further analysis in this context, such as other feature extraction techniques like wavelet transforms and other classifiers like neural networks or SWLDA. It is likely that there are interaction effects between all of these choices.

Overall, the contributions of this thesis are the comparison of multiple BSS methods, some of which have not been applied to this problem before, and the comparison of several feature selection algorithms that have been adapted for source component selection and channel selection. All possible combinations are explored to analyze any interaction effects between algorithms. The experiments are run on three different datasets, two subjects from the BCI Competition III dataset [11] and one subject recorded from the Colorado State University Occupational Therapy lab. A comparison of these approaches across multiple subjects is seldom found in the current literature of the BCI field. Therefore, this work will contribute to the overall understanding of the effect of different preprocessing steps when used for P300 classification.

1.4 Overview

The thesis is laid out as follows: Chapter 2 defines all algorithms that are used in the experiments for this work. The three BSS algorithms, time-delay embedding, and SVMs are all mathematically defined, while each feature selection algorithm is described in detail based on how they are applied in this work. Chapter 3 describes the datasets used and details the steps used in all experiments, which can be followed in order to reproduce any of the results. Chapter 4 explains the results of all experiments and includes figures that show the comparison of performances between different approaches. Chapter 5 concludes this work by providing a more in-depth discussion of the results and several different avenues for future work.

Chapter 2

Algorithms

In order to better understand the different approaches that were used in the experiments, a detailed explanation of all algorithms is provided in this chapter. Support vector machines (SVMs) are first described to understand how the classification of target/non-target trials is applied in this context. The general approach towards all BSS algorithms is first described before detailing the differences between the three specific algorithms used in this work. Then, the process of creating time-delay embedded data from the original dataset is described, and it is explained how this relates to BSS since, in this work, time-delay embedding is only ever used by the BSS algorithms. Finally, the four feature selection algorithms are described based on how they are used for either component selection or channel selection.

2.1 Support Vector Machines

A support vector machine (SVM) [39] is a large-margin binary classifier. Given a set of training data from two classes, an SVM will find the hyperplane that separates samples from the two classes with a maximum margin. The training set for an SVM consists of paired instances x_i with a class label y_i. A training set containing N instances with f features is represented by $\{(x_i, y_i)\ |x_i \in \mathbb{R}^f, y_i \in \{-1, 1\}\}_{i=1}^{N}$. If the data is linearly separable, the optimization problem can simply be formulated as

$$\begin{aligned}&\min_{w,b} \|w\|^2\\ &\text{s.t.}\ \ y_i(w \cdot x_i + b) \geq 1\end{aligned}$$

with the separating hyperplane designated by $w \cdot x + b = 0$. When an unseen instance x from the test set is classified, the decision function is $f(x) = w \cdot x + b$. If $f(x) < 0$, the instance lies on one side of the hyperplane and x is classified as belonging to the first class (i.e., $y = -1$), and if $f(x) > 0$, x is in the second class (i.e., $y = 1$).

If the data is not linearly separable, the formulation can be modified to become a soft-margin classifier. Misclassifications are now allowed with a given penalty that is regulated by the penalty parameter C that must be chosen in advance. The optimization problem is now stated as

$$\begin{aligned} &\min_{w,b} \|w\|^2 + C \sum_{i=1}^{N} \xi_i^2 \\ &\text{s.t.} \;\; y_i(w \cdot x_i) \geq 1 - \xi_i, \;\; \xi_i \geq 0. \end{aligned}$$

The variable ξ_i can be seen as a slack variable. The soft-margin formulation finds a hyperplane solution that is a trade-off between maximizing the margin and the amount of error allowed with a given penalty parameter C.

The above formulations, only allow for linear classification, but these can be extended by applying the kernel trick for nonlinear classification. A data instance x can be mapped into some higher-dimensional (or even infinite-dimensional) space by a given mapping $x \mapsto \Phi(x)$. If only the dot product between two instances is needed, which is the case in the dual formulation of the SVM, the kernel trick can be used so that the mapping is never explicitly computed for any instances. A kernel function $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ is defined that efficiently computes the equivalent of the dot product of x_i and x_j in the higher-dimensional space. In this work, a Gaussian radial basis kernel is used that is defined as

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2). \tag{2.1}$$

Based on optimization theory, the SVM optimization problem can be reformulated as its dual problem. The dual is an equivalent optimization problem that is formulated using Lagrangian multipliers. The SVM optimization problem in its dual form is

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) && (2.2) \\ &\text{s.t.} \;\; \sum_{i=1}^{N} \alpha_i y_i = 0, \;\; \alpha_i \geq 0. \end{aligned}$$

The decision function is $f(x) = \sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b$, which still classifies an instance x depending on whether the value of $f(x)$ is positive or negative.

In this context, the SVM is utilized by training it with target and non-target instances using feature vectors created from the time samples of the signals from a one second window in each of the channels being using. The implementation used for this work is from the R package *e1071* [40].

2.2 Blind Source Separation

Blind source separation (BSS) methods are based on the assumption that the observed signals from a multi-channel recording are produced from a mixture of several distinct source signals. They attempt to isolate the original source signals by applying a transform to the set of observed signals. The classic example used to describe BSS methods is the cocktail party problem. In the "cocktail party," there is a room with several people speaking simultaneously, and their voices are considered to be the independent sources in this problem. There are also several microphones set up throughout the room recording a mixture of all the voices (sources). Using the information from all of the recording channels (microphones), the BSS methods attempt to transform this data into a set of the original independent source signals.

The BSS methods accomplish this by finding a set of basis vectors to transform the original data, where the three different methods use different optimization criteria to find these basis vectors. The observed signals $X(t) = [x_1(t),\ x_2(t),\ x_3(t),\ \dots,\ x_n(t)]$ consisting of n channels are assumed to originate from some unknown linear mixing of p source signals $S(t) = [s_1(t),\ s_2(t),\ s_3(t),\ \dots,\ s_p(t)]$, defined as

$$X(t) = AS(t).$$

Here, A is the unknown mixing matrix of dimensions $n \times p$. It should be noted that in order to solve this problem, the mixing matrix must be overdetermined (i.e., $n \geq p$). If there are more sources than recording channels, it is impossible to completely isolate each source. In this work, an assumption is made for simplification that the number of sources is equal to the number of channels, $p = n$. Therefore, the mixing matrix A is $n \times n$. The BSS method then attempts to find the corresponding unmixing matrix W that best estimates the original source signals, as in

$$\hat{S}(t) = WX(t).$$

Here, $\hat{S}(t)$ is the estimation of the source components $S(t)$. It is also possible to find an estimate of the original mixing matrix A by taking an inverse of the resulting unmixing matrix, as in W^{-1}. Using the mixing matrix A, it is then possible to project the components back to reconstruct the original data, shown by

$$X(t) = W^{-1}\hat{S}(t) = A\hat{S}(t) = AWX(t).$$

It is also possible to select only a single source component $\hat{s}_j(t)$ and project it back to find the contribution from component j. $X^{(j)}(t)$ denotes the signals in $X(t)$ that originated from the source

$\hat{s}_j(t)$ and is defined as

$$X^{(j)}(t) = a_j \hat{s}_j(t) = a_j w_j X(t). \tag{2.3}$$

The variable a_j refers to the j^{th} row in A, and w_j refers to the j^{th} column in W.

The unmixing matrix $W = [w_1,\ w_2,\ w_3,\ \ldots,\ w_n]$ consists of a set of basis vectors, where each column vector w_i is associated with a single source, and can be thought of as a spatial filter. Each basis vector provides weights for a linear mixture of the channels from the recorded data $X(t)$ that results in a single source component. This is especially relevant when it is known that the sources are spatially distributed across the recording channels. Therefore, there will be many references to the spatial filters associated with sources, which are derived from the associated basis vector. Figure 2.1 provides a visual example of an extracted source and its associated spatial filter from an 8-channel EEG recording.

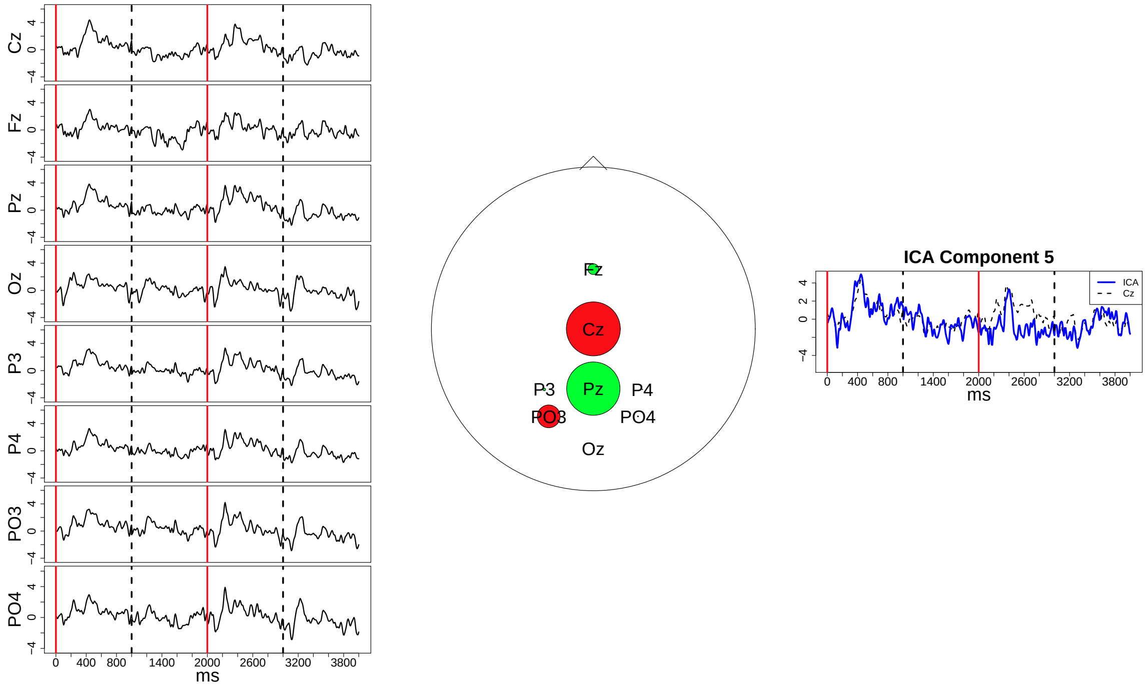


Figure 2.1: An example ICA source component from Subject C. The original 8 channels are shown on the left with alternating target and non-target trials. The start of a target trial is indicated with a red solid line. The scalp map visualizes the spatial filter associated with this component. The sign of the weight is indicated by the color: green (light gray) for positive, and red (dark gray) for negative. The magnitude of the weight is relative to the size of the circle. The linear combination of the original 8 channels using the given spatial filter produces the single source component as shown on the right in blue. This source component contains activity associated with the positive peak in P300 trials and is overlaid on the original signal from electrode Cz.

2.2.1 Principal Component Analysis

Principal component analysis (PCA) [41] is a method to find a linear transformation of the data that maximizes the variance of the transformed data. Using PCA, the transformation is constrained to be orthogonal. This problem is solved using the eigenvectors of the covariance matrix $X^T X$, as shown in

$$X^T X \alpha = \lambda \alpha$$

where α is an eigenvector with a corresponding eigenvalue λ. The set of all eigenvectors constitute the new orthogonal basis that the data is projected on.

PCA can be used for BSS on EEG data recorded with n channels by applying it to the covariance matrix across the n dimensions. The bases obtained from this process are linear combinations of the n channels, which represent the set of spatial filters for the source components.

2.2.2 Independent Component Analysis

Independent component analysis (ICA) is a concept that can be applied to any set of random variables to find a linear transform that maximizes the statistical independence of the output components. Whereas PCA relies only on the second-order statistics of covariance, ICA can be seen as a higher-order generalization of PCA that does not require orthogonality. Comon [42] rigorously defined ICA as an optimization problem to minimize the mutual information between the source components. He presented an efficient algorithm using higher-order statistics to measure the notion of non-Gaussianity that corresponds with statistical independence.

To understand how non-Gaussianity relates to statistical independence, it is necessary to understand the central limit theory, which states that the sum of many independent processes tends towards a Gaussian distribution. Therefore, if $S(t)$ is assumed to be a set of truly independent sources, the observed mixed signal $X(t)$ will be more Gaussian by the central limit theory. A single estimated source $\hat{s}_i(t)$ is a linear mixture of $X(t)$ given by the weights in the spatial filter w_i. The w_i that maximizes the non-Gaussianity of $\hat{s}_i(t)$ is used to find the closest approximation to the true independent source $s_i(t)$. Therefore, the optimization criteria is to find the unmixing matrix that maximizes non-Gaussianity in all of the source components.

There are many different implementations of ICA that each use different metrics for statistical independence. Based on the comparison of several ICA algorithms applied to BCI applications by Kachenoura, et al., [43], we chose to use FastICA [44], utilizing the R implementation in the package *fastICA* [45]. FastICA uses kurtosis as a measure of non-Gaussianity, which is a fourthorder statistic defined as $E\{X^4\} - 3$ for signals with a mean of zero and unit variance. It indicates whether a probability distribution is sub-Gaussian, Gaussian, or super-Gaussian. A super-Gaussian distribution contains a more acute peak around the mean and long tails. It is usually the result of a signal that does not normally vary much but contains large infrequent deviations. A sub-Gaussian distribution is represented by a lower, wider peak. An assumption is made that white noise is Gaussian, and the most independent sources are furthest from a Gaussian distribution. A signal with negative kurtosis is sub-Gaussian, while a signal with positive kurtosis is super-Gaussian, and a Gaussian distribution has a kurtosis equal to zero. A signal that is super-Gaussian with a higher kurtosis means that much of the variation in the signal is caused by large deviations that occur infrequently causing a heavy tailed distribution that is more common in many independent signals. FastICA finds the set of source components that correspond to the kurtosis extrema.

2.2.3 Maximum Noise Fraction

Maximum noise fraction (MNF) is an algorithm that was first proposed to eliminate noise from satellite images. It was later adapted as a BSS technique for time series data by Hundley and Kirby [46, 47]. It attempts to decompose the signal into source and noise components, based on the assumption that the observed signal X is created by a combination of sources S and noise N, as in $X = S + N$. An assumption must also be made that the sources and noise are orthogonal, that $S^T N = 0$ and $N^T S = 0$. The algorithm then finds the basis vectors α that maximize the SNR, where SNR is defined as

$$SNR = \max_{\alpha \neq 0} \frac{\|S\alpha\|}{\|N\alpha\|} = \max_{\alpha \neq 0} \frac{\alpha^T S^T S\alpha}{\alpha^T N^T N\alpha}$$

With the assumption that the sources and noise are orthogonal, the following equality holds

$$\frac{\alpha^T X^T X\alpha}{\alpha^T N^T N\alpha} = \frac{\alpha^T (S+N)^T (S+N)\alpha}{\alpha^T N^T N\alpha} = \frac{\alpha^T S^T S\alpha}{\alpha^T N^T N\alpha} + \frac{\alpha^T N^T N\alpha}{\alpha^T N^T N\alpha} = \frac{\alpha^T S^T S\alpha}{\alpha^T N^T N\alpha} + 1$$

Therefore, the optimization equation can be rewritten as

$$SNR = \max_{\alpha \neq 0} \frac{\alpha^T S^T S\alpha}{\alpha^T N^T N\alpha} = \max_{\alpha \neq 0} \frac{\alpha^T X^T X\alpha}{\alpha^T N^T N\alpha} \tag{2.4}$$

Since $X^T X$ is easily computed with the input data X, only the noise covariance $N^T N$ is unknown and must be characterized. If the source signals are smooth, the noise covariance can be approximated as the covariance of the difference between the original signal X with the same signal shifted by one time sample X_S, so that $N^T N \approx (X - X_S)^T (X - X_S)$. This characterizes the noise as large fluctuations from one sample to the next, which generally corresponds with the higher frequency components. Equation 2.4 can be solved using the generalized singular value decomposition to find the basis vectors α, resulting in the source components that are then sorted in order of decreasing SNR.

2.3 Time-delay Embedded Data

Time-delay embedded data is created by adding a new lagged dimension using the same data from one of the original channels, only shifted (or lagged) by one or more time samples. The original recorded data $X(t)$ with n channels is represented as

$$X(t) = \left[\begin{array}{cccc} x_1(t), & x_2(t), & \dots, & x_n(t) \end{array} \right]$$

where $x_i(t)$ represents a column vector of all time samples recorded from channel i starting at time 0. The data can then be shifted by τ time samples so that $x_i(t+\tau)$ represents a column vector of time samples from channel i starting at time τ instead of time 0.

$$X(t+\tau) = \left[\begin{array}{cccc} x_1(t+\tau), & x_2(t+\tau), & \dots, & x_n(t+\tau) \end{array} \right]$$

In order to create the modified time-delay embedded data matrix $X^*(t)$, d new lagged dimensions are added from each of the original channels, creating a new matrix that is nd-dimensional from the original n dimensions.

$$X^*(t) = \left[\begin{array}{ccccc} X(t), & X(t+\tau), & X(t+2\tau), & \dots, & X(t+d\tau) \end{array} \right] \tag{2.5}$$

This effectively adds temporal information to each individual time sample. In the original data matrix $X(t)$, a single point at time t contains one time sample from each of the n channels, therefore encompassing purely spatial information. The corresponding point in $X^*(t)$ contains the same spatial information from the n channels as well as a window of temporal information between time t and time $t + d\tau$. With this additional temporal information, the BSS methods result in spatio-temporal filters that have the potential to extract temporal correlations between source activity in separate locations of the brain. Figure 2.2 displays a simple example that shows the advantages of temporal information. There are four sources that are each represented by sine waves at a specific frequency. However, the first two sources are the same frequency, only with a phase shift. The four sources are mixed together into three recording channels. When using spatial BSS, the algorithms do not find any correlation between the phase-delayed sources and are unable to extract independent sources using three channels. However, when only a single lagged dimension is included in the data, both ICA and MNF are clearly able to extract components that have the same frequencies as the original sources. The spatio-temporal BSS algorithms result in six components since there are six dimensions including the lags. With the temporal information, the two phase-delayed sources can be thought of as originating from a single source that causes temporally correlated but spatially distinct activity. PCA was not able to extract the independent sources, most likely due to the orthogonal constraint on the basis vectors.

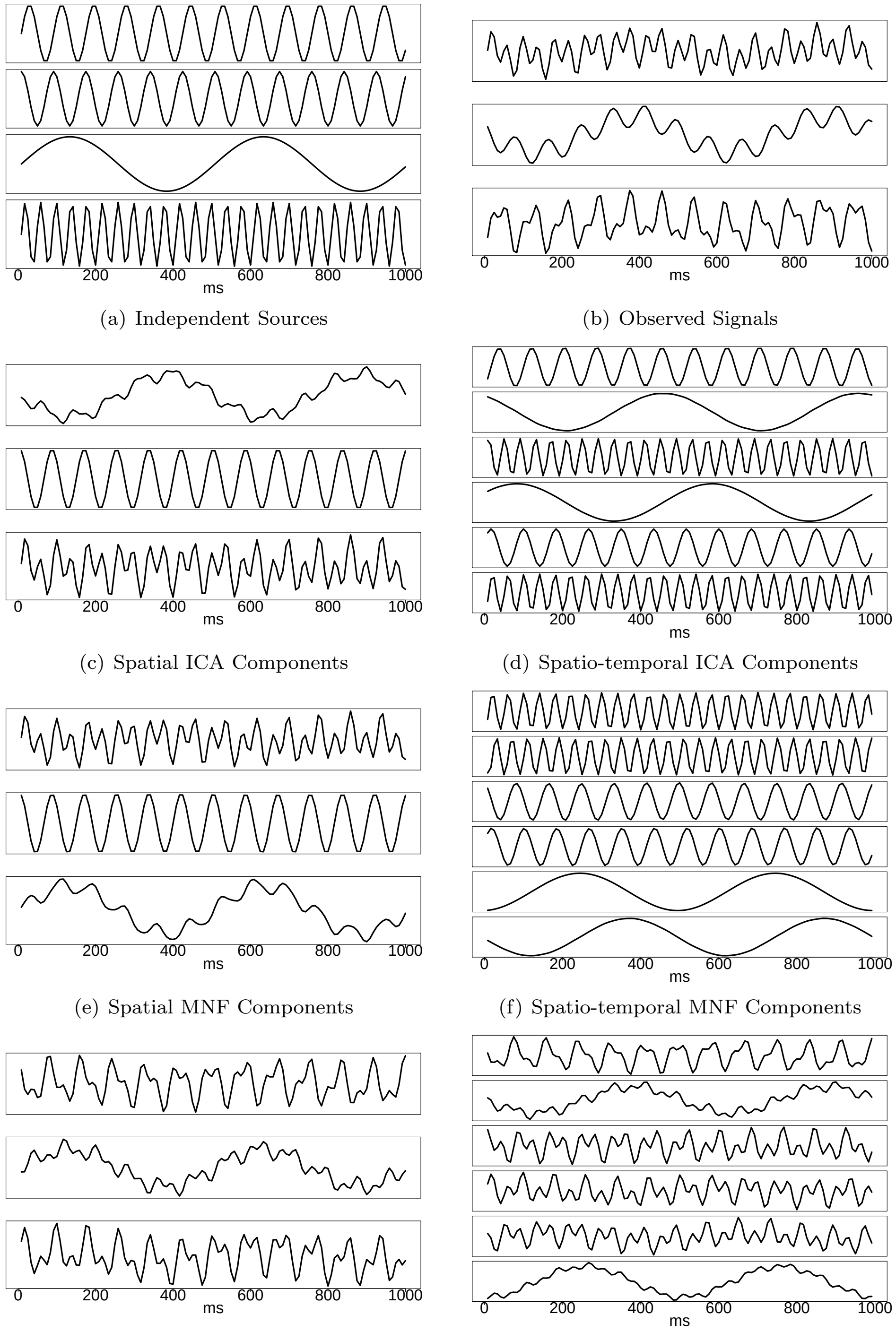
2.4 Component Selection

When applying BSS, the objective is to minimize the mutual information across the channels, resulting in source components that can capture very specific brain activity. The set of source components can contain some signals that are relevant for classification as well as signals that are irrelevant. Figure 2.1 shows a relevant source component that contains a signal capturing the positive P300 peak. However, in the same set of components, one source extracted only the visual response to the stimuli that is found in both target and non-target trials, as shown in Figure 2.3. Since this activity is common among all trials, it does not provide any distinguishing information for the classifier, and therefore, it is believed that the application of a component selection step can eliminate these irrelevant components.

The following feature selection algorithms were adapted for source component selection, but they can also be used identically for channel selection on the original data. When applied to the original data without BSS, certain electrodes can be entirely eliminated because they are not useful for P300 detection. Each algorithm returns a ranking of the components (or channels) by relevance. In addition to the ranking, it is necessary to determine the optimal number of components. Using a separate validation set, the top m ranked components were selected, as m varied from 1 to a preset maximum of 20, to find the m that resulted in the highest validation accuracy using an SVM. Thus, the top m ranked components were selected to be used for the final data transform.

2.4.1 Forward Selection

Forward selection is a greedy algorithm that is popular for feature selection [48]. It attempts to find the best features by analyzing the performance of certain subsets of features using the same learning algorithm that is later used for classification, in this case an SVM. Starting with an empty set, each component was added, and the accuracy was assessed. The component with the highest validation accuracy was selected and added to the current set. New components were recursively added to the set in this fashion, and the components were ranked based on the order they were added.



(a) Independent Sources

(b) Observed Signals

(c) Spatial ICA Components

(d) Spatio-temporal ICA Components

(e) Spatial MNF Components

(f) Spatio-temporal MNF Components

(g) Spatial PCA Components

(h) Spatio-temporal PCA Components

Figure 2.2: Comparison of the three BSS methods with and without time-delay embedding. The four original sources (a) are sine waves at different frequencies, except that the first two are the same frequency with a $\frac{\pi}{2}$ phase shift. The four sources are artificially mixed together into three channels that represent the observed recordings (b). The source components are shown (c)–(h) for each BSS method as the standard spatial algorithm and as the spatio-temporal version when applied to time-delay embedded data containing one lag.

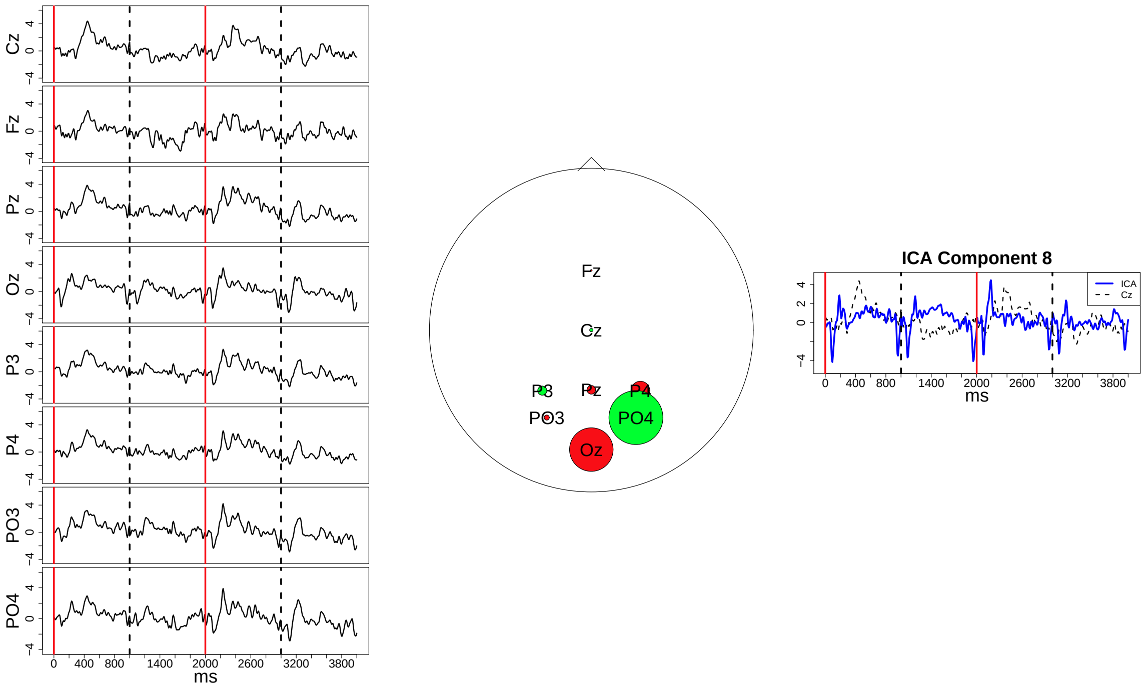


Figure 2.3: An example ICA source component from Subject C. The original 8 channels are shown on the left with alternating target and non-target trials. The start of a target trial is indicated with a red solid line. The scalp map visualizes the spatial filter associated with this component. The sign of the weight is indicated by the color: green (light gray) for positive, and red (dark gray) for negative. The magnitude of the weight is relative to the size of the circle. The linear combination of the original 8 channels using the given spatial filter produces the single source component as shown on the right in blue. This source component contains activity associated with the visual processing of the flashing stimuli that occurs around 100–200 ms in both the target and non-target trials. As seen in the spatial filter, this component contains a large contribution from the back of the head over the occipital lobe where visual information is processed in the brain.

2.4.2 ANOVA Ranking

This algorithm is based on the analysis of variance (ANOVA) statistical test, which has been utilized previously for feature selection [49]. The ANOVA test was applied to individual features using trials from both classes in order to determine the level of significant difference between the means of each class. Since each feature represents a single time sample associated with one component, the ANOVA scores were averaged over all time samples for each component. The averaged ANOVA scores were then used to rank the components.

2.4.3 Relief

Relief was originally developed by Kira and Rendell [36] as an efficient heuristic estimator of feature weights. In this case, the features are individual time samples in the signals, so the algorithm was adapted for component selection by averaging the feature weights across all time samples for a given component. Using a set of training instances from both classes (target and non-target), a random sample is chosen, and the two nearest neighbors from each class are found, the nearest hit (same class) and the nearest miss (opposite class). The weights for each feature are updated based on rewarding short distances to the nearest hit and longer distances to the nearest miss. The adapted Relief implementation in these experiments uses the L_1 distance measure and the five nearest neighbors from each class based on an extension proposed by Kononenko [50]. The components were ranked based on their averaged feature weights.

2.4.4 Recursive Feature Elimination

RFE is tied closely with the learned weights from an SVM model. It removes features with the smallest weights since they have the least influence on the margin and recursively retrains the SVM to remove the next set of weights. Lal et al. [31] adapted this for reverse channel elimination, and the same concept is used here by averaging the SVM model weights across each component. At each iteration, the bottom $\frac{1}{4}$ of the components with the smallest weights were removed due to the large amount of computational resources required for this algorithm. The final components are ranked based on their averaged weights.

Chapter 3

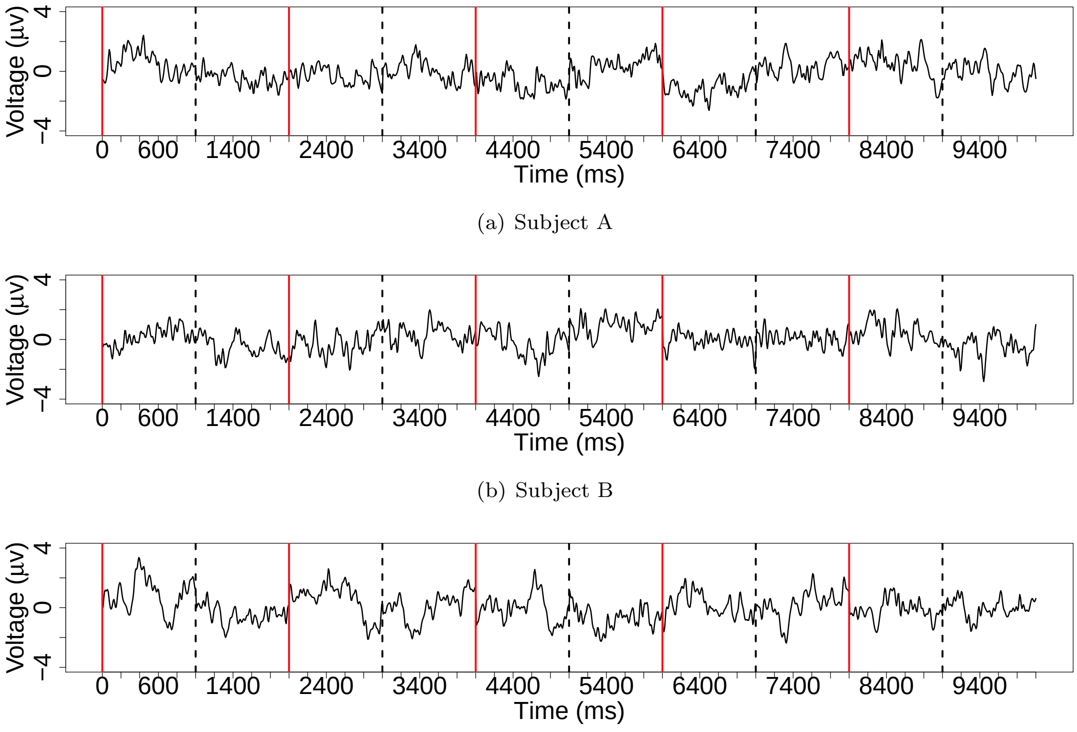
Experimental Methods

This chapter describes in detail the steps used to generate all results in this thesis. The datasets from all three subjects are first described. Then, a step-by-step description of the experimental flow is provided. Some steps are optional, which are noted, but all possible steps based on the application of BSS and/or feature selection are described from beginning to end. All parameter choices are explained based on the results of several pilot experiments. Then, the hypotheses are explicitly stated while describing the set of experiments that were run in order to test these hypotheses.

3.1 Datasets

The first dataset is taken from the BCI Competition III, dataset II [11] that contains responses obtained using the P300 speller paradigm as described by Farwell and Donchin [8]. There are two subjects in the dataset, referred to as Subject A and Subject B, containing 2550 target trials each. The data were originally sampled at 240 Hz and decimated by a factor of 2. The data were recorded using a 64-electrode cap, but some of the experiments use only a subset of 8 channels (Fz, Cz, Pz, Oz, P3, P4, PO7, PO8). Krusienski, et al., [37] empirically showed that this subset of electrodes achieved the best performance across many subjects.

The second dataset was recorded at the Colorado State University Occupational Therapy lab using the Biosemi active electrode system. The paradigm for this study consisted of a grid of flashing letters with only the letter in the middle of the screen changing with each flash. The subject was asked to focus their attention on the middle letter and count the occurrences of a specific target letter ("b", "d", or "p"). The probability of occurrence for the target letter was 0.25. The data were originally sampled at 1024 Hz and decimated by a factor of 8. The data were recorded with a 32-electrode cap, but again, some of the experiments use only 8 channels (Fz, Cz, Pz, Oz, P3, P4, PO3, PO4). 540 target trials were recorded from a single subject, a 23-year-old able-bodied female, referred to as Subject C. Figure 3.1 shows several trials of the training data for each subject.



(a) Subject A

(b) Subject B

(c) Subject C

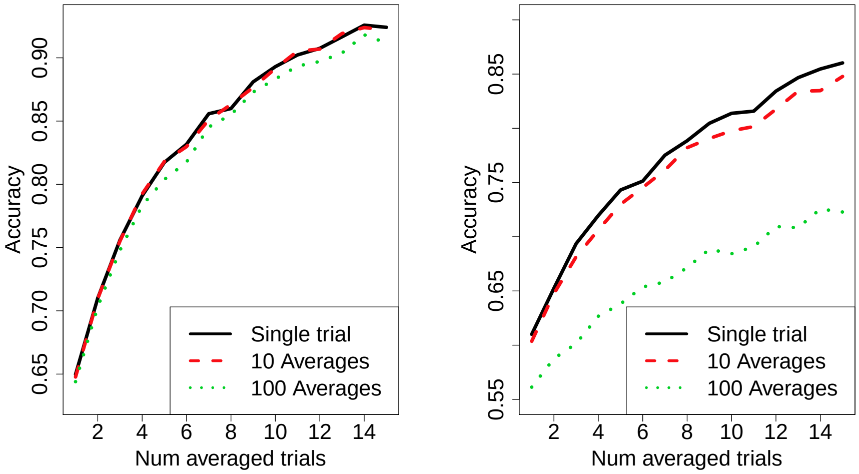
Figure 3.1: Training data for each subject. 5 individual trials are averaged together for one of the trials shown here, which is the same that is used to train the BSS algorithm and the SVM. The target and non-target trials are alternately concatenated, with a red solid line indicating a target trial and a black dashed line indicating a non-target trial.

3.2 Work Flow

The entire work flow described below details the steps taken for a single experiment with a given approach and was performed independently for each subject. The output is a set of preprocessing steps and an SVM model used for future classification. Using data from only a single subject, the algorithm can adapt to the individual differences in each subject's P300 signal, allowing for better classification results on the same subject. This is the standard practice for most BCI systems, requiring each subject to first undergo a training session before the BCI system can be used.

All data were first bandpass filtered from 0.23 Hz to 30 Hz. The data were then normalized for zero mean and unit variance. This was especially necessary for the BCI Competition data since each character in the sequence was a separate data segment containing 15 repetitions of the flashing of all 12 rows/columns, and some individual data segments had much larger amplitudes. Therefore, each data segment was normalized independently, and each channel was also treated independently. The trials used for classification consist of one-second long windows after each stimulus onset that are extracted from the continuous signal in each data segment. If time-delay embedded data was utilized though, the additional lagged dimensions were added before the data was segmented into one-second trials. Therefore, using n channels and d lags, the original n-dimensional continuous signal becomes a nd-dimensional time-delay embedded data matrix that is then used for extracting the one-second trials. As explained above, the time-delay embedded data allows the BSS methods to extract spatio-temporal filters than can account for more complex brain dynamics. However, in the experiments in which BSS was not applied, there is no reason to include additional lagged dimensions because it does not add any new data. Without applying BSS, the feature vector used for the SVM is produced from all time samples in each of the channels. If time-delay embedded data was used, this would only result in redundant features since there would be identical time samples in the lagged dimensions. Therefore, when later discussing the results of time-delay embedding, it is only used in the context of BSS.

A subset of the non-target trials was randomly selected to produce a balanced dataset with the same amount of target and non-target trials. All trials were then randomly partitioned into training, validation, and test data, using dataset fractions of 0.3, 0.3, and 0.4, respectively. At this point, the training set consists of an equal number of *individual* target and non-target trials. It is possible to average together groups of these trials to produce training data that has a cleaner P300 signal as shown in Figure 1.5, but it is not immediately clear whether this is beneficial. Some initial pilot experiments were run to test the effects of averaging trials together in the data used to train the BSS algorithm as shown in Figure 3.2. At first it might seem that having a more reliable and consistent P300 signal in the training data would allow the BSS method to extract more meaningful components. For instance, Figures 2.1 and 2.3 were produced using 100 averaged trials because the larger number of averaged trials produced source components that were easily reproducible and visually looked more meaningful. However, when used for P300 classification, the number of averaged trials in the test data usually only varies from 1 to 15. Therefore, it seems that the components obtained from a BSS method on training data with fewer averaged trials does indeed carry over better to test data using fewer averaged trials. In all remaining experiments, the number of averaged trials applied to the BSS training data was fixed at 5 averaged trials, which was arbitrarily chosen based on these results. After the training data was averaged together, target and non-target trials were alternately concatenated together to produce the training data used for BSS. Since class labels are not used for BSS methods, the training data essentially consisted of one continuous signal for each channel containing one-second windows of alternating P300 and non-P300 responses, as shown in Figure 3.1.



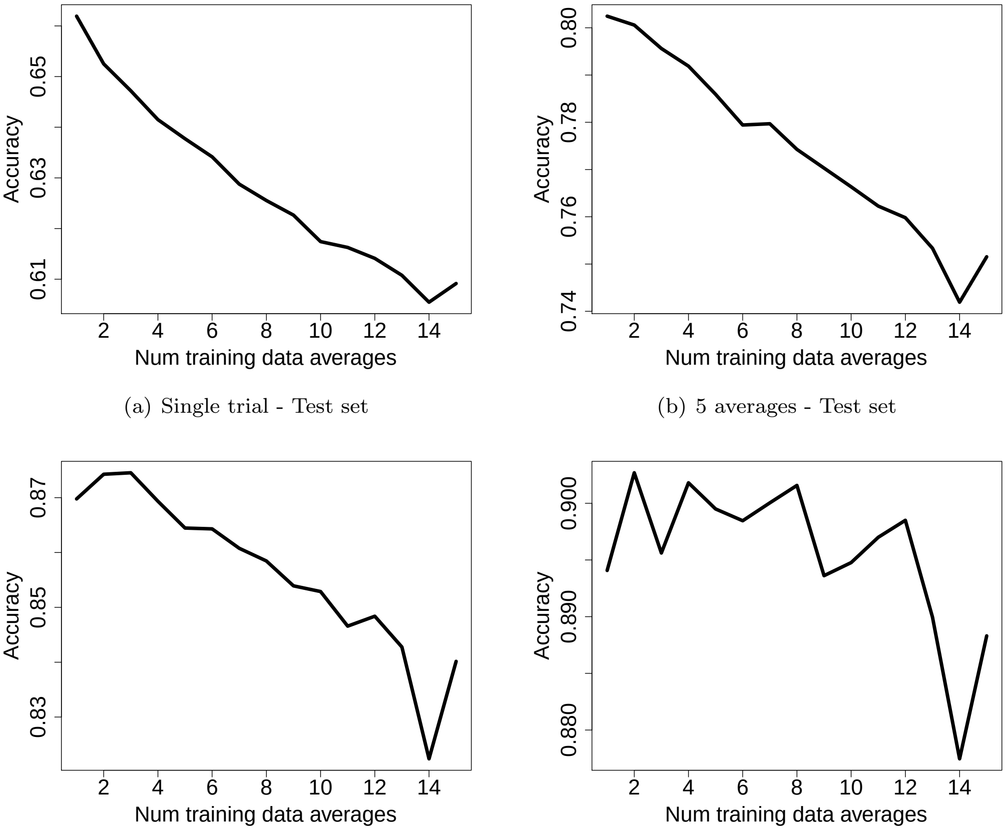
(a) 8 channels

(b) 64 channels

Figure 3.2: Results of changing the number of trials averaged together for the training data for BSS. Results are from Subject A when using ICA without any feature selection. The classification accuracies are shown as the number of averaged trials in the test data varies from 1 to 15. Three different experiments were run with the number of averaged trials set to 1, 10, or 100, which only applies to the training data used for ICA. These results show that a smaller number of averaged trials applied to the training data allows ICA to extract more meaningful source components used for classification.

If one of the BSS methods was used, the transform matrix was applied to both the training and validation data. At this point, feature selection was an optional step to find the final set of features used to train the SVM. For the best performance of the feature selection algorithms, they should be applied to the same type of data that will be used for training the SVM. The number of trials averaged together for this training data is a separate parameter than the number of averaged trials used for the BSS training data. Again, this parameter will have an effect on the performance, and it is unclear what the optimal number of averaged trials is. As more trials are averaged together, the training data becomes less noisy, but it also reduces the number of instances in the training set. Figure 3.3 shows the effects of increasing the number of averaged trials used for the SVM training data. As can be seen, the effect is different depending on the number of averaged trials used on the test set. Since a more general classifier is desired that performs well on test data with a varying number of averaged trials, the parameter was fixed for 5 averaged trials on the training data for the

SVM.



(a) Single trial - Test set

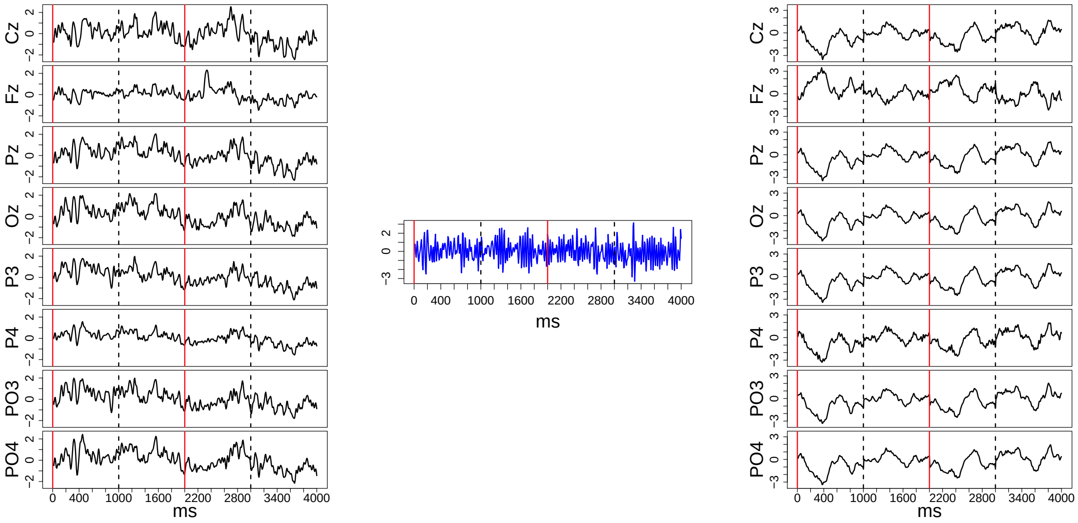
(b) 5 averages - Test set

(c) 10 averages - Test set

(d) 15 averages - Test set

Figure 3.3: Results from Subject A of changing the number of trials averaged together for the training data for the SVM. The four plots show the results when the number of averaged trials used on the test set is 1, 5, 10, and 15.

A special step was required if time-delay embedded data was used. Figure 3.4 shows a source component obtained from ICA with a spatio-temporal filter applied to the time-delay embedded data. It is clear to see that the source component itself is hard to interpret, but once it is projected back to the original data space using Equation 2.3, it is more meaningful. The projection shows the contribution of this individual component to the original 8-channel recording. If all components were projected back and summed together, it would reconstitute the original data exactly. Therefore, all components were projected back to the original data space, but it was necessary to keep them separate for the feature selection step. This greatly increases the memory requirements and computational demand of this step. As an example, an 8-channel recording with 9 additional lagged dimensions would produce 80 source components. Since each individual component is projected back to the original 8 channels and kept separate, this produces 640 dimensions in the projected data space that the feature selection algorithms operate on. For n channels and d lags, the number of dimensions required for this step is $(d+1) \cdot n^2$. Because of this, the time-delay embedding experiments were only run on the 8-channel data.



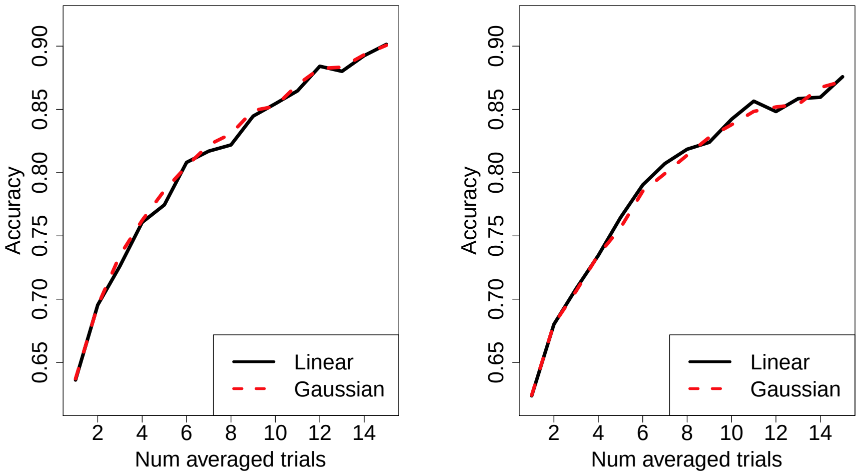
(a) Original 8 channels (b) Source component (c) Projected component

Figure 3.4: Results from Subject C of ICA applied to 8-channel data with 10 lags. The original 8 channels are shown in (a). After the lags were added to this data, ICA was applied, and one of the resulting components is shown in (b). The single component is projected back to the original data space showing its contribution to the 8 recorded channels as shown in (c).

Once the training and validation data was averaged together (and projected back if necessary), feature selection was applied. Using one of the algorithms defined in Section 2.4, it was either applied for component selection (if BSS was used) or channel selection (if BSS was not used). The algorithm was applied to the training data to find the weights for each channel/component to rank them. After the channels/components are ranked, it is still necessary to find the optimal number of channels/components to use. The validation dataset was used for this purpose. As m varied from 1 to 20, the top m ranked components were selected, and an SVM was trained on the training data, and the validation accuracy was found on the validation dataset. The m that was associated with the highest validation accuracy signified the optimal number of channels/components to use. In all experiments, the maximum number of components allowed was set to 20. This proved to be sufficient in most cases because the optimal number was usually below 20. If time-delay embedding was used for BSS, and the components were all projected back, the final set of optimal components was summed together in the original data space.

The final training data for the SVM was generated by applying the BSS transform (if applicable), averaging 5 trials together, and selecting the final set of selected components/channels. If no BSS was applied, the feature vector was created from all time samples in the final set of selected channels. If BSS was performed without time-delay embedding, the feature vector was created from all time samples in the final set of source component signals, and if BSS was performed on time-delay embedded data, the final set of components are projected back to the original data space and summed together, and the feature vector was created from all time samples from the projected signals using all n channels.

A Gaussian kernel is used for the SVM in all experiments. Figure 3.5 shows a comparison between using an SVM with a linear kernel and an SVM with a Gaussian kernel. There is no significant difference between the choice of kernels in this case. Using a Gaussian SVM, there are two parameters that must be selected: the cost penalty C, and the γ parameter in the Gaussian kernel function defined in Equation 2.1. For each experiment, these two parameters were tuned through a grid search of all possible parameter combinations of $C = \{10, 100, 1000\}$ and $\gamma = \{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ using cross-validation on the training set.



(a) 8 channels

(b) 64 channels

Figure 3.5: Results comparing a linear SVM with an SVM using a Gaussian kernel. Results are shown for Subject A using a subset of 8 channels (a) and the full set of channels (b).

To generate the test data, the same BSS transform was applied (if used), and the same set of components/channels was selected. In order to assess how well the classifier performs with different numbers of averaged trials, it is standard practice to assess the accuracy as the number of averaged trials varies, in this case from 1 to 15. Therefore, the same test set was reused for single trial classification, 2 averaged trials, 3 averaged trials, etc.

3.3 Experiment Design

The first hypothesis was that the application of a BSS method would transform the data to contain more meaningful information, and that the application of feature selection would allow for the elimination of any irrelevant or noisy source components. To test this, all possible combinations of BSS methods and feature selection algorithms were run. With the possibility of three BSS methods (or not using one) and four feature selection algorithms (or not using one), there are 20 possible approaches that were compared. These experiments were run on 8-channel data and the full set of channels in order to understand the differences based on the number of electrodes available. Although an 8-channel system is much more practical and more desirable to use in a patient's home, there is still much research being performed that uses the full set of electrodes.

The second hypothesis was that the addition of temporal information for BSS through time-delay embedded data would allow for more complex spatio-temporal patterns to be extracted from the brain activity. The temporal information has the ability to capture phase-delayed coherence between electrodes that can result in a more reliable indicator of P300 source activity. Therefore, it was hypothesized that BSS performed on time-delay embedded data would result in better performance for P300 classification over using purely spatial information. In order to test this, the classification accuracies were assessed as the amount of temporal information was varied by changing the number of lagged dimensions d, as shown in Equation 2.5. Experiments were run with $d = 0$, which is the same as purely spatial information, up to $d = 15$. Also, the effect of the number of time samples between each lagged dimension was analyzed by varying the parameter τ. Experiments were run with a fixed d while τ varied from 1 to 20.

The performance metric used was the classification accuracy using a Gaussian SVM. To obtain the final results shown for each approach, we ran 10 repetitions with different random partitions, ensuring that the same partition was used when comparing between approaches. It was also controlled to make sure that a different set of random partitions was used for the final results than was used for choosing any of the parameters described above in Section 3.2. Many existing approaches report their accuracies on character classification of the BCI Competition III dataset by grouping the responses together from the same rows and columns and determining the target row and the target column. However, only the effects of binary classification on target and non-target trials are studied here. While the binary classification accuracies can be used successfully to assess the performance of a classifier, they cannot be directly compared to the results in the literature. However, by analyzing only binary classification results, this allows for a consistent comparison between datasets, since Subject C was not recorded using the P300 speller paradigm with flashing rows/columns.

Chapter 4

Results

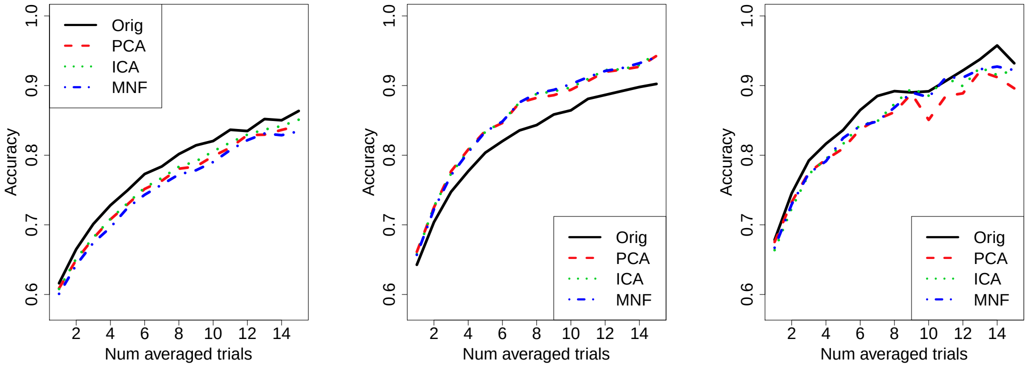
The results of the experiments that were described above are presented in this chapter. While the observations of the performance trends are noted with the results, a more in-depth discussion is saved for the next chapter. First, the results are shown from using the full set of electrodes, and then the results are shown from running the same experiments on the subset of 8 electrodes. Finally, the last section describes the results that explore the performance of time-delay embedded data.

4.1 Comparison of BSS Methods

Although experiments were run on all possible combinations of BSS methods and feature selection algorithms for each subject using the full set of electrodes and a subset of 8 electrodes, it is not possible to visually display all the results here. A summary of all approaches is shown in Table 4.6 averaged across all subjects, and the individual results for each subject are included in the Appendix. The results of the comparisons between approaches were analyzed with a two-way ANOVA model for a fixed subject and fixed number of averaged trials (10). Tests of pairwise comparisons of the average accuracies between methods were considered, and Tukey's method was used to control the experiment-wise error rate.

4.1.1 Full Set of Electrodes

The following results are from the experiments run on the full set of electrodes, which correspond to 64 channels in Subjects A and B and 32 channels in Subject C. To first analyze the effects of BSS without using any feature selection, Figure 4.1 shows each of the three methods applied to each subject's data. The accuracies of the original data (using no BSS or feature selection) are always shown with a solid black line as a reference. The first observation is that BSS shows different results depending on the subject. The accuracy is decreased in Subjects A and C, but all three BSS algorithms significantly improve the performance in Subject B ($p < 0.001$). Immediately, it is clear how hard it is to generalize the results of different approaches across many subjects. In all subjects, the results from the three BSS methods are comparable to each other.



(a) Subject A

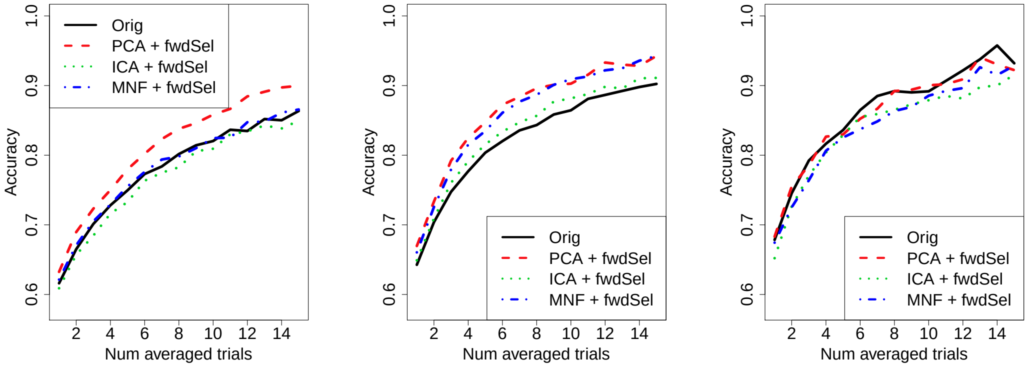
(b) Subject B

(c) Subject C

Figure 4.1: Comparison of the three BSS methods applied to the full set of electrodes for each subject.

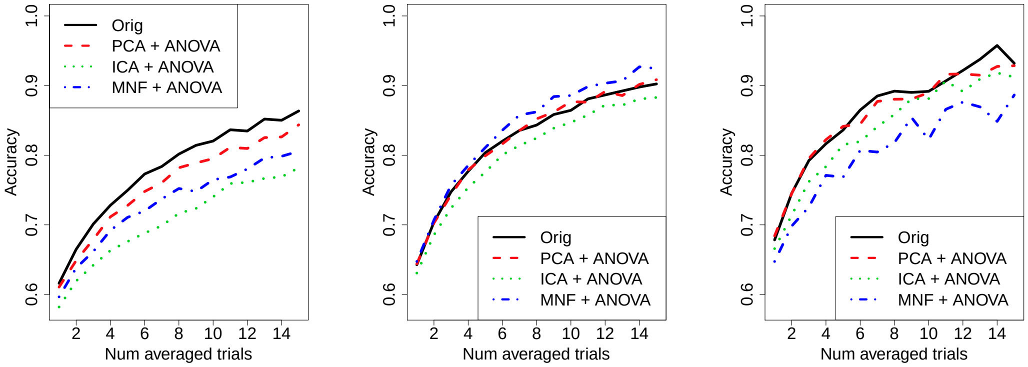
The application of BSS to the full set of electrodes results in 64 (or 32) source components. As stated above, the application of feature selection can eliminate many of the irrelevant source components. Figures 4.2, 4.3, 4.4, 4.5 show the results of applying forward selection, ANOVA ranking, Relief, and RFE, respectively, after applying BSS. Depending on the subject and the BSS algorithm, the feature selection step generally increases the performance, except for when using ANOVA ranking. The largest performance gain comes from applying RFE. There are also interesting trends that can be seen in the relative rankings of the three different BSS methods. Although there is hardly any difference between the methods when applying only BSS as in Figure 4.1, the feature selection step causes more separation in the relative performances. Again, it is not consistent between subjects, but surprisingly, the simplest method, PCA, performs the best in most cases. This is seen especially in Subject A where PCA is significantly better than ICA in all cases ($p < 0.01$). MNF is also a simple method based on second-order statistics that performs well in many of the cases on Subjects A and B. However, it is interesting to note that the relative ranking of MNF and ICA are reversed in Subjects B and C. ICA is the most complex algorithm that utilizes higher-order statistics, but there is only one experiment (Subject C in Figure 4.3) where ICA is significantly better than either of the other two methods, in this case MNF ($p < 0.05$).

When RFE is applied after BSS as in Figure 4.5, the performance is increased over the baseline in all subjects. However, in order to better understand whether the performance increase was due



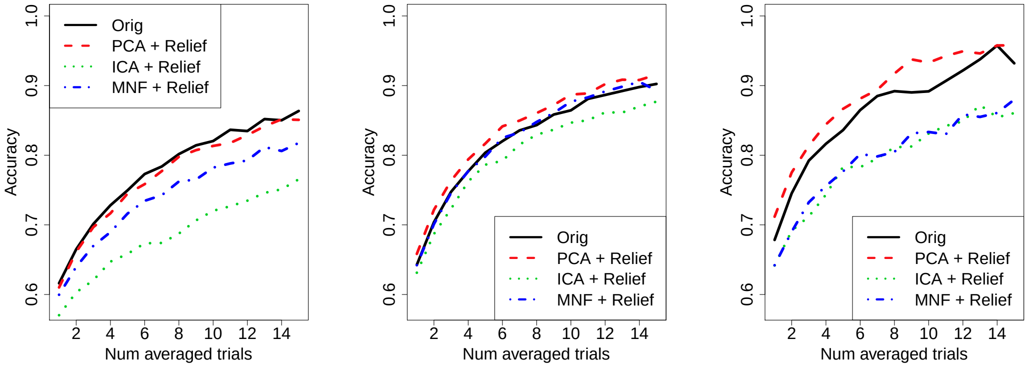
(a) Subject A (b) Subject B (c) Subject C

Figure 4.2: Comparison of the three BSS methods as applied to the full set of electrodes when using forward selection to select the relevant components.



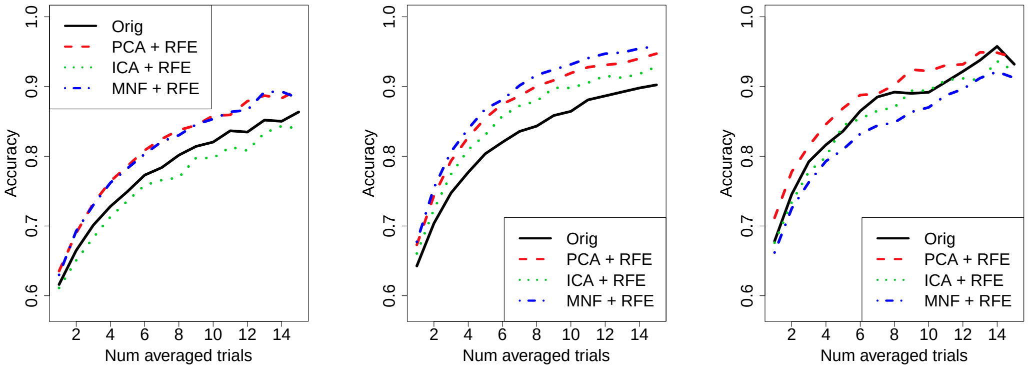
(a) Subject A (b) Subject B (c) Subject C

Figure 4.3: Comparison of the three BSS methods as applied to the full set of electrodes when using ANOVA ranking to select the relevant components.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.4: Comparison of the three BSS methods as applied to the full set of electrodes when using Relief to select the relevant components.



(a) Subject A

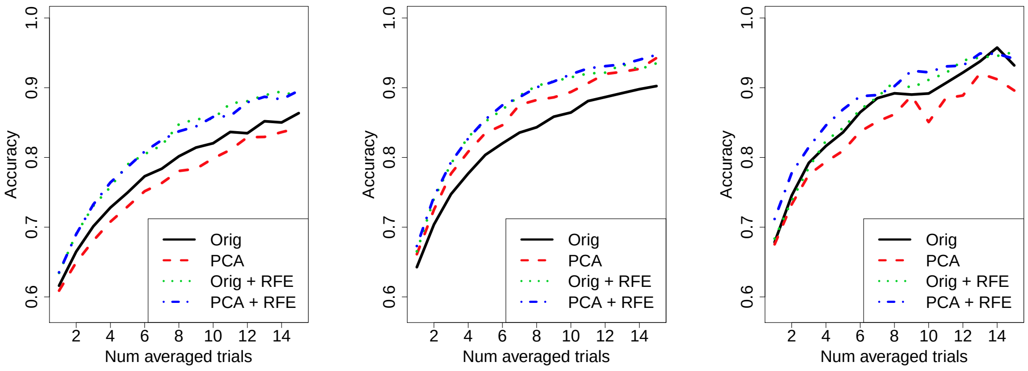
(b) Subject B

(c) Subject C

Figure 4.5: Comparison of the three BSS methods as applied to the full set of electrodes when using RFE to select the relevant components.

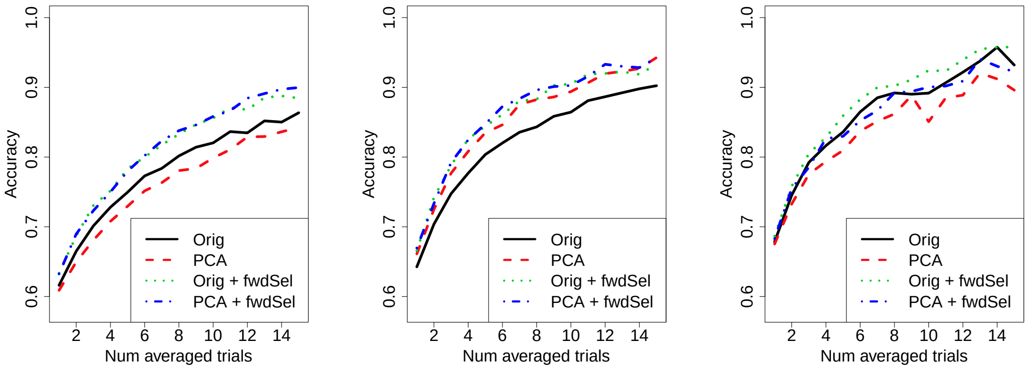
to the BSS method or feature selection, Figure 4.6 shows the results of PCA used in conjunction with RFE compared to the results of each used in isolation. RFE provides a significant performance boost in Subject A ($p < 0.05$) and Subject B ($p < 0.001$) whether or not PCA is applied beforehand. Similar results can be seen in Figure 4.7 that shows the same comparison for forward selection. In each case, the performance of only applying feature selection is the same or better than the other two approaches. One exception is Subject C in Figure 4.6 that suggests a slight improvement from the application of PCA and RFE combined, but it is not significant. Figure 4.8 shows the same comparison for MNF and RFE that displays similar trends with another exception of Subject B, where the combination of MNF and RFE results in a slightly higher performance gain, but again, it is not significant. The same trends are seen for ICA as well, except that the performance is even lower, especially in Subject A, as seen in Figure 4.9. Only forward selection and RFE were mentioned in these experiments because when analyzing ANOVA ranking and Relief, the trends are inconsistent, but it generally decreases the accuracy even further whether applied to the original data or to the BSS components. Other than the exceptions mentioned above, the combination of BSS with feature selection does not provide any additional advantages over the application of only the feature selection algorithm. Therefore, when using a full electrode array, the main cause for performance gains comes from channel selection.

Since it is seen that the application of BSS does not provide any additional advantages, the feature selection algorithms are further analyzed in the context of channel selection on the original data. Since each algorithm returns a ranking of the channels by relevance, it is possible to ignore the optimal number of channels returned and use the top m ranked channels instead, as m varies



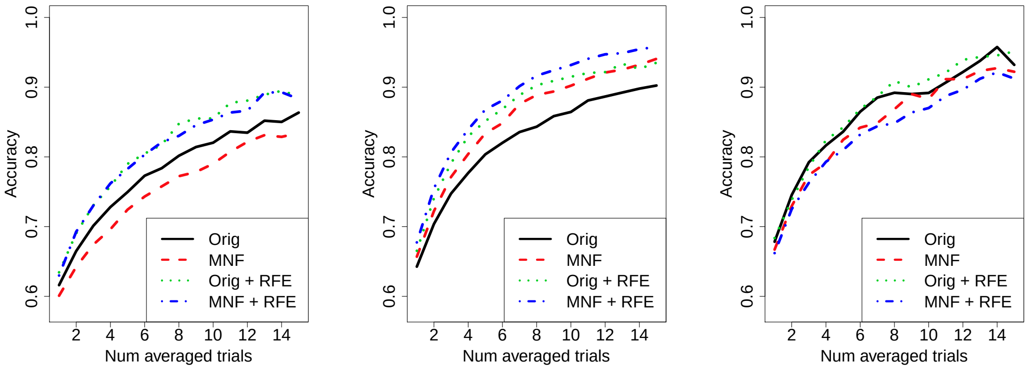
(a) Subject A (b) Subject B (c) Subject C

Figure 4.6: Comparison of PCA using RFE for feature selection along with the results of PCA and RFE each independently applied to the original data. Results are from the full set of electrodes.



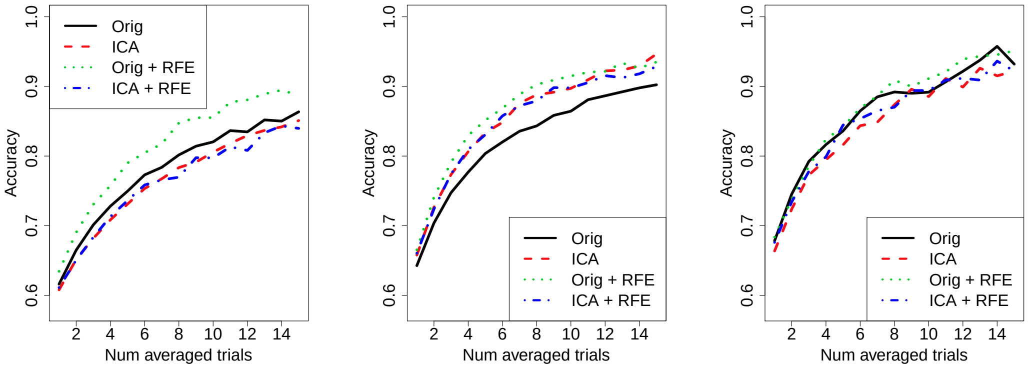
(a) Subject A (b) Subject B (c) Subject C

Figure 4.7: Comparison of PCA using forward selection to select the relevant components along with the results of PCA and forward selection each independently applied to the original data. Results are from the full set of electrodes.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.8: Comparison of MNF using RFE for feature selection along with the results of MNF and RFE each independently applied to the original data. Results are from the full set of electrodes.



(a) Subject A

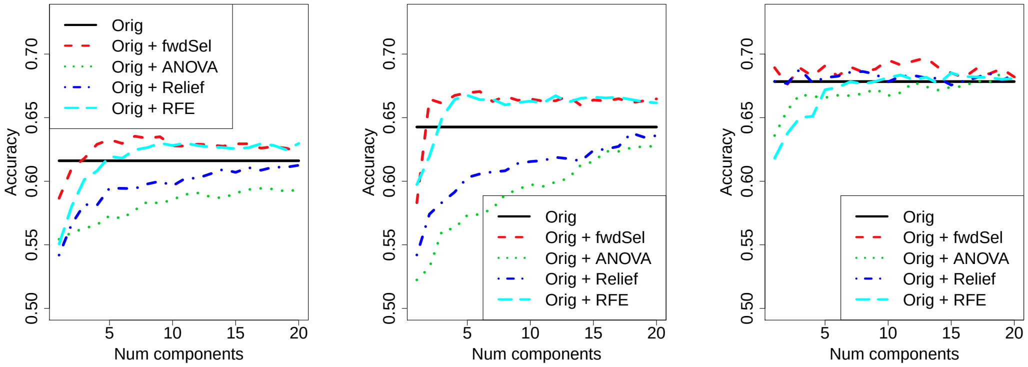
(b) Subject B

(c) Subject C

Figure 4.9: Comparison of ICA using RFE for feature selection along with the results of ICA and RFE each independently applied to the original data. Results are from the full set of electrodes.

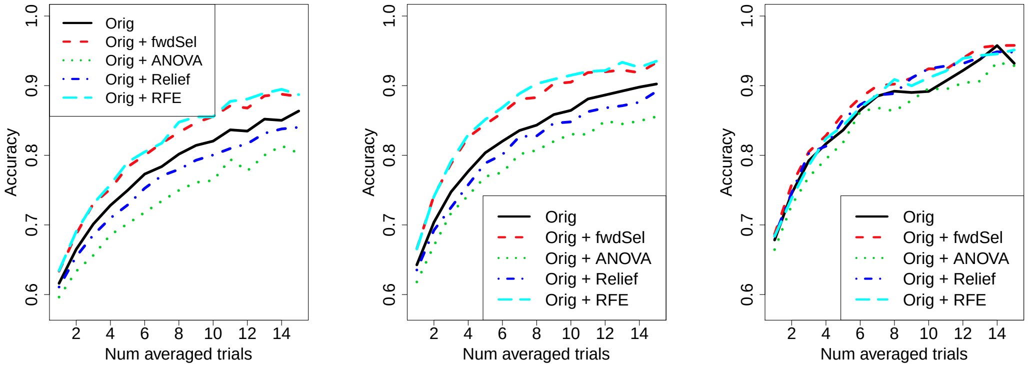
from 1 to 20. This provides better insight into the relative channel ranking of each algorithm, which is shown in Figure 4.10. The accuracies are shown here only for single trial classification, but the trend stays the same as the number of averaged trials increases. Although there is not much of a difference between the algorithms in Subject C, forward selection and RFE are clearly better in Subjects A and B. ANOVA ranking and Relief do not perform better than the baseline using the top 20 ranked channels, but forward selection and RFE are able to exceed the baseline performance with only 2–5 channels. In Subject C, the same accuracy can be achieved using only the single best channel. It is interesting to note that although 64 (or 32) channels of data are available, there are only a few channels that are necessary for classification, and the elimination of irrelevant channels increases the performance. Figure 4.11 shows the same results using the optimal number of channels returned from each algorithm while varying the number of averaged trials applied to the test data.

In general, these results show that both forward selection and RFE are better algorithms to use for channel selection. The optimal number of channels for each algorithm is listed in Table 4.1, and this shows that RFE is able to achieve good performance while still selecting the fewest number of channels. Tables 4.2, 4.3, and 4.4 show the average channel ranking (over 10 repetitions) for each feature selection algorithm for Subjects A, B, and C, respectively. First, it is clear to see that different channels are ranked higher depending on the subject, indicating that the spatial distribution of each subject's P300 signal is unique. Also, forward selection and RFE produce rankings that are more similar to each other as opposed to ANOVA ranking and Relief, which is why they show similar performance.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.10: Comparison of feature selection algorithms used for channel selection on the full set of electrodes. Accuracies are shown for single trial classification as the number of top ranked channels varies from 1 to 20. The results from the original data are flat because all channels are always being used.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.11: Comparison of feature selection algorithms used for channel selection on all electrodes when the optimal number of channels is used.

Table 4.1: Average number of optimal channels chosen by each feature selection algorithm for each subject when applied to all electrodes.

Algorithm	Subject A	Subject B	Subject C
Forward selection	13.45	11.00	8.09
ANOVA ranking	14.81	13.36	9.72
Relief	13.36	15.64	6.45
RFE	12.00	8.91	6.45

Table 4.2: The top 8 ranked channels (out of all possible channels) for Subject A.

Algorithm	Subject A							
Forward selection	Po7	Pz	Cpz	P7	Fc1	Cz	Po8	Fc5
ANOVA	F1	Fc1	Fcz	T10	C6	Fc4	F3	Af4
Relief	Fc1	Cz	Oz	Fz	Cpz	C4	F1	C6
RFE	Po7	Po8	Cpz	Pz	P7	Cz	Iz	P5

Table 4.3: The top 8 ranked channels (out of all possible channels) for Subject B.

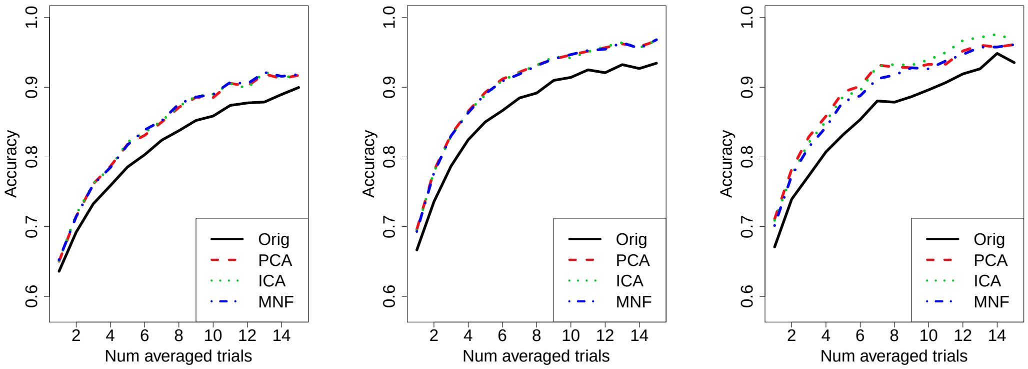
Algorithm	Subject B							
Forward selection	Po8	Cpz	Po7	Pz	O1	Cz	C6	Cp3
ANOVA	Po8	F7	Ft8	T10	C6	F6	Fc5	Af3
Relief	Po8	P8	Fc5	Fpz	P6	F8	F7	Af7
RFE	Po8	Cpz	O1	Po7	Cz	Pz	P8	Po4

Table 4.4: The top 8 ranked channels (out of all possible channels) for Subject C.

Algorithm	Subject C							
Forward selection	T7	CP1	FC1	Cz	FP1	C3	P7	F7
ANOVA	FC1	CP1	F4	Cz	Pz	P3	C4	T7
Relief	Cz	T7	CP1	Pz	C4	FC1	CP2	P3
RFE	Cz	T8	T7	CP1	P7	C3	P8	P3

4.1.2 Subset of Electrodes

The same set of experiments was also run on the subset of 8 electrodes for each subject as specified in Section 3.1. This is important because it represents a smaller set of electrodes for a more practical system that can be used in a patient's home. Figure 4.12 shows the results of applying each of the BSS methods to the 8-channel data. In contrast to the full set of electrodes, the results from this experiment show that BSS significantly improves the accuracy in all subjects: Subject A ($p < 0.001$), Subject B ($p < 0.0001$), and Subject C ($p < 0.05$). Also, each of the three methods performs equally well.

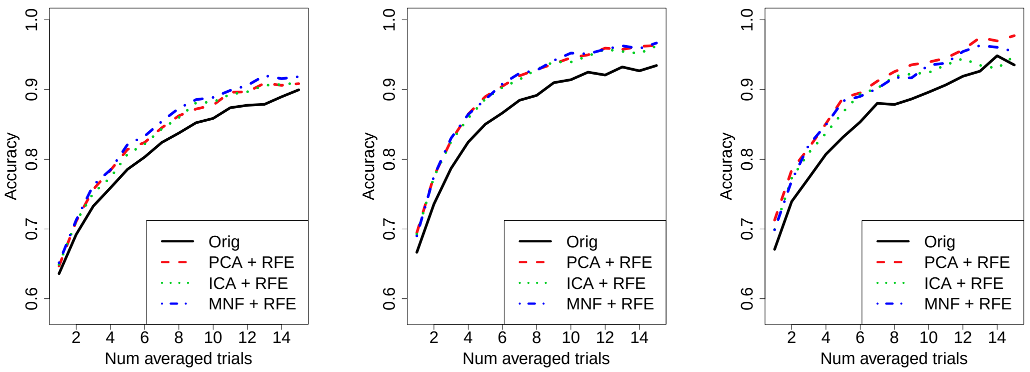


(a) Subject A (b) Subject B (c) Subject C

Figure 4.12: Comparison of the BSS methods without component selection on the 8-channel data.

Figure 4.13 shows the results of the same BSS transforms, except with the application of RFE for component selection. The performance is almost unchanged, and in a few cases, the performance degrades slightly. Only RFE is shown here since it has already proven to be one of the better feature selection algorithms. However, the results from the other feature selection algorithms were still analyzed and found to be almost identical. It is clear that the elimination of components after BSS provides no additional advantages.

Again, it is necessary to analyze whether the performance gains are due to BSS or the feature selection step. Since both PCA and RFE have already been found to perform well, Figure 4.14 compares the results of using PCA in conjunction with RFE along with the results of each step applied independently. This shows that the application of RFE does not make much of a difference whether used for component selection after PCA or for channel selection on the original data. When RFE is used for channel selection, the results are almost the same as the baseline, except for a slight increase in Subject C that is not significant. The results are similar for all combinations of



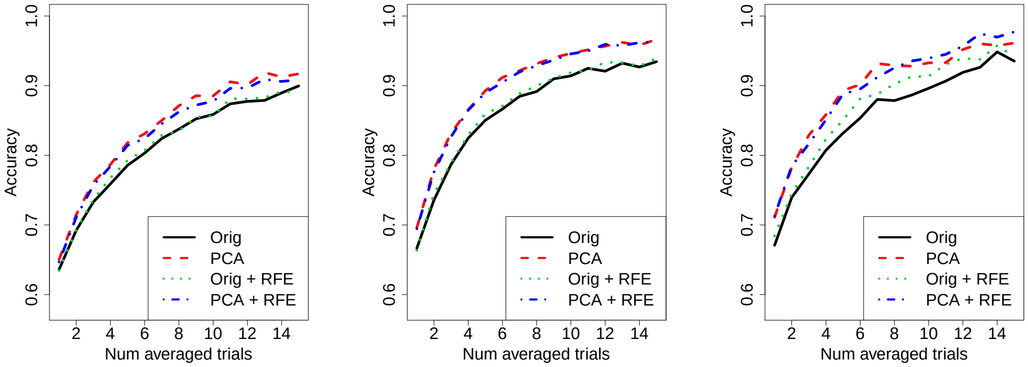
(a) Subject A

(b) Subject B

(c) Subject C

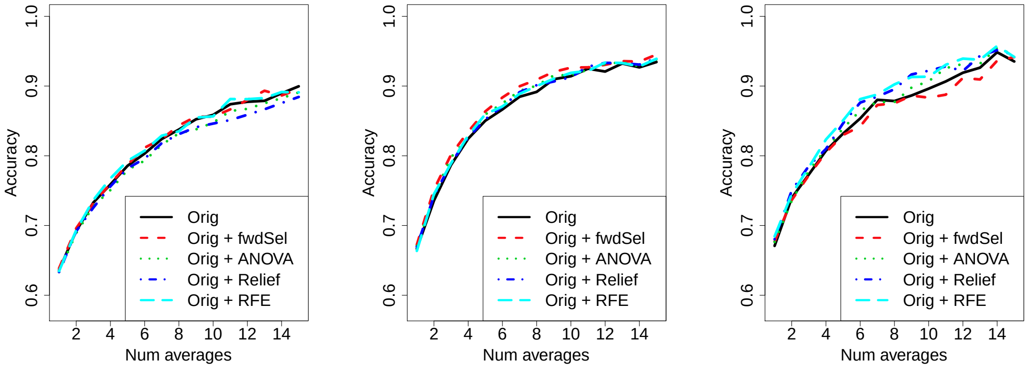
Figure 4.13: Comparison of the BSS methods using RFE for component selection on the 8 channel data.

BSS methods and feature selection algorithms, and therefore, they are not shown here. To further show that feature selection does not significantly change the performance, Figure 4.15 shows the application of all feature selection algorithms that result in a similar performance to the baseline. With only 8 available channels, it might first seem that no channels are being eliminated since they are all known to be relevant. However, Table 4.5 shows that the feature selection algorithms are indeed eliminating channels, but they do not result in any change in performance. This set of experiments on the 8-channel data shows opposite effects as those run on the full electrode set in regards to using BSS or feature selection. Whereas feature selection is the major contributor to the performance gain when using all electrodes, it has no significant effect on the 8-channel data, and BSS is shown to be the cause of the performance gain. To better understand the differences between subjects and between BSS algorithms Figures 4.16, 4.17, and 4.18 show a comparison of the averaged top ranked component (over 100 repetitions) for each BSS method for Subjects A, B, and C, respectively. For each repetition, the single best component was selected with forward selection. In general, across all three subjects, the three algorithms produce similar components that contain a strong contribution from the top of the head (electrodes Cz and Pz). The only method that is distinctly different is PCA applied to Subject A because instead, it shows a strong contribution from Fz. However, even with this difference, PCA still performed similarly to the other methods in Subject A.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.14: Comparison of PCA using RFE for feature selection along with the results of PCA and RFE each independently applied to the original data. Results are from the 8-channel data.

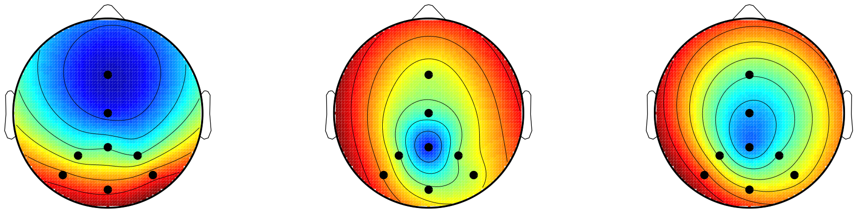


(a) Subject A (b) Subject B (c) Subject C

Figure 4.15: Comparison of feature selection algorithms applied for channel selection on the 8-channel data.

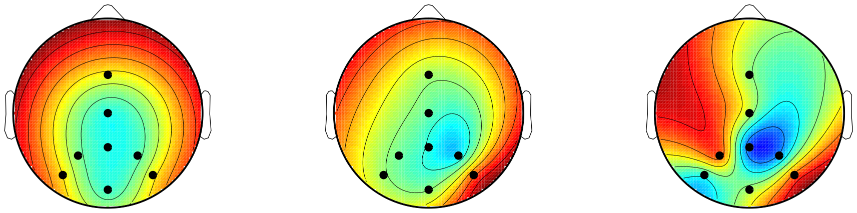
Table 4.5: Average number of optimal channels chosen by each feature selection algorithm for each subject when applied to the 8-channel data.

Algorithm	Subject A	Subject B	Subject C
Forward selection	4.81	3.72	2.09
ANOVA ranking	7.54	6.18	3.45
Relief	7.27	6.27	2.18
RFE	4.81	4.36	3.18



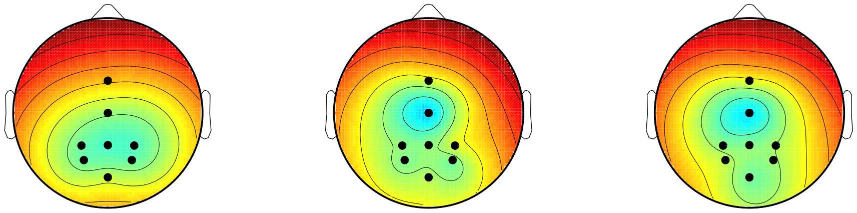
(a) PCA (b) ICA (c) MNF

Figure 4.16: The average top ranked component from each of the BSS methods for Subject A.



(a) PCA (b) ICA (c) MNF

Figure 4.17: The average top ranked component from each of the BSS methods for Subject B.



(a) PCA (b) ICA (c) MNF

Figure 4.18: The average top ranked component from each of the BSS methods for Subject C.

Table 4.6: Classification accuracy for each possible combination of BSS and feature selection algorithms. The mean and standard deviation across all three subjects are shown when using 8 channels and when using all channels. The accuracies for each individual subject are provided in the Appendix.

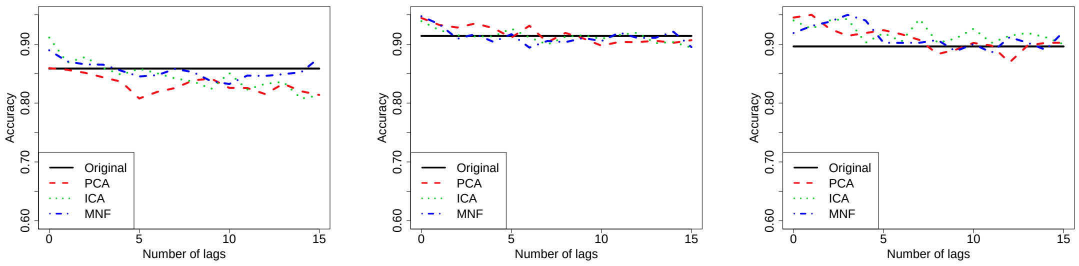
BSS	Feat sel	Using 8 Channels			Using All Channels		
		1 avg	8 avg	15 avg	1 avg	8 avg	15 avg
Orig	none	.66 ± .016	.87 ± .026	.92 ± .026	.65 ± .017	.85 ± .031	.90 ± .027
ICA	none	.68 ± .020	.91 ± .023	.95 ± .024	.64 ± .016	.85 ± .032	.91 ± .040
MNF	none	.68 ± .020	.91 ± .027	.95 ± .030	.64 ± .017	.84 ± .035	.90 ± .037
PCA	none	.69 ± .017	.91 ± .026	.95 ± .028	.65 ± .016	.84 ± .037	.89 ± .037
Orig	fwdSel	.66 ± .021	.88 ± .032	.93 ± .032	.66 ± .019	.87 ± .034	.93 ± .031
ICA	fwdSel	.68 ± .020	.91 ± .037	.95 ± .028	.64 ± .022	.83 ± .045	.89 ± .046
MNF	fwdSel	.68 ± .020	.89 ± .039	.94 ± .034	.65 ± .022	.85 ± .042	.91 ± .041
PCA	fwdSel	.68 ± .021	.90 ± .037	.94 ± .038	.66 ± .022	.88 ± .038	.92 ± .034
Orig	ANOVA	.66 ± .020	.87 ± .025	.92 ± .032	.63 ± .022	.81 ± .052	.86 ± .062
ICA	ANOVA	.68 ± .024	.91 ± .024	.95 ± .022	.63 ± .028	.80 ± .049	.86 ± .050
MNF	ANOVA	.68 ± .020	.91 ± .027	.95 ± .025	.63 ± .028	.81 ± .048	.87 ± .051
PCA	ANOVA	.68 ± .021	.90 ± .021	.95 ± .024	.65 ± .019	.84 ± .038	.89 ± .048
Orig	Relief	.66 ± .018	.88 ± .035	.92 ± .032	.64 ± .028	.83 ± .044	.89 ± .046
ICA	Relief	.68 ± .020	.91 ± .021	.95 ± .021	.61 ± .024	.78 ± .051	.83 ± .055
MNF	Relief	.68 ± .018	.90 ± .027	.95 ± .031	.63 ± .025	.80 ± .045	.86 ± .053
PCA	Relief	.68 ± .019	.91 ± .030	.94 ± .026	.66 ± .017	.86 ± .029	.91 ± .034
Orig	RFE	.66 ± .020	.88 ± .030	.92 ± .039	.66 ± .015	.89 ± .028	.92 ± .029
ICA	RFE	.68 ± .019	.90 ± .030	.94 ± .026	.65 ± .020	.84 ± .032	.90 ± .031
MNF	RFE	.68 ± .020	.91 ± .023	.95 ± .029	.66 ± .019	.87 ± .033	.92 ± .039
PCA	RFE	.68 ± .024	.91 ± .033	.95 ± .027	.67 ± .019	.88 ± .032	.93 ± .030

4.2 Analysis of Time-delay Embedding

In this section, the experiments will analyze the effects of the amount of temporal information included (the number of lags d) and the nature of this information (the number of time samples τ between the lagged dimensions). Again, there are many interaction effects from the choice of the BSS method and the choice of the feature selection algorithm, so the experiments were run on all combinations with the exception of RFE. When applied to the high dimensional data after including lags, RFE proved to be too computationally intensive, and there were not enough resources available to run the full set of experiments with RFE. In the previous section, the accuracy was assessed as the number of averaged trials used on the test set varied from 1 to 15. In this section, although the results were computed and analyzed in the same way, the figures only show the results using 10 averaged trials since it was found that the performance trends are not dependent on the number of averaged trials used on the test set.

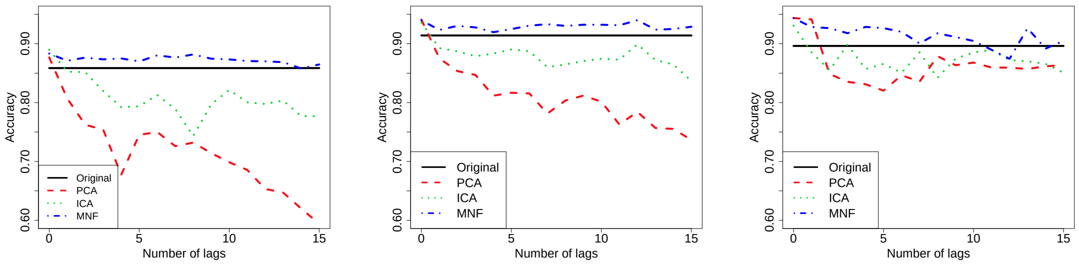
The first set of experiments tests the effects of varying the number of lags d from 0 to 15 with $\tau = 1$. Remember that the feature selection step is necessary when using time-delay embedded data because if no components are eliminated, all components would be projected back and summed together to reconstitute the original data. Therefore, the effects of BSS applied to time-delay embedded data cannot be assessed independent of the feature selection step. Figures 4.19, 4.20, and 4.21 show the accuracies when component selection is performed by forward selection, ANOVA ranking, and Relief, respectively. The baseline accuracy is a flat line because no lags are ever used in the original data. When $d = 0$, the approach is the same as spatial BSS as described in the previous section. In all cases, as more lags are added, the performance of the algorithm drops or stays level.

In general, when using forward selection or Relief, the two feature selection algorithms have similar performance and maintain accuracies around the baseline. Also, all three BSS methods show similar performance. However, the results for ANOVA ranking in Figure 4.20 show an interaction effect that clearly separates the three BSS methods. It shows the best performance with MNF since the accuracy stays slightly above the baseline. However, with PCA, the performance is extremely poor, dropping down to an accuracy of 0.60 in Subject A. When comparing the components selected by each feature selection algorithm, it is clear that forward selection and Relief both choose components towards the beginning of the list of PCA components, but ANOVA almost exclusively selects components from the end. Since the components from PCA are ordered based on how much variance they capture, this means that ANOVA ranking is only selecting components that capture a very small amount of variance in the original data.



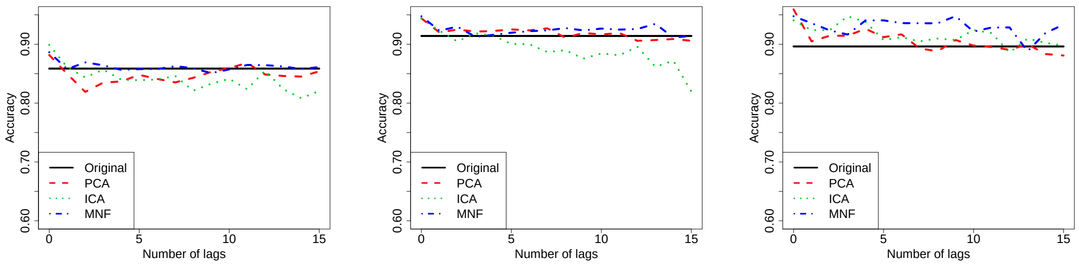
(a) Subject A (b) Subject B (c) Subject C

Figure 4.19: Comparison of BSS methods applied to time-delay embedded data as the number of lags are varied using forward selection for component selection.



(a) Subject A (b) Subject B (c) Subject C

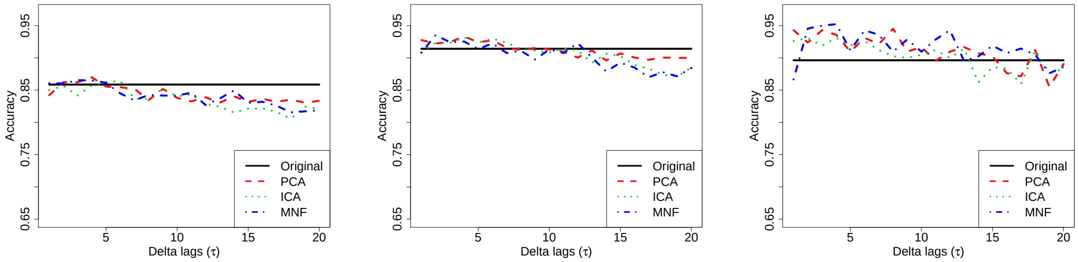
Figure 4.20: Comparison of BSS methods applied to time-delay embedded data as the number of lags are varied using ANOVA ranking for component selection.



(a) Subject A (b) Subject B (c) Subject C

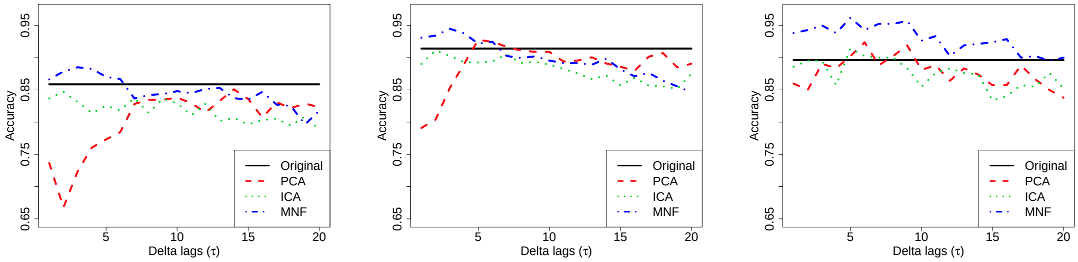
Figure 4.21: Comparison of BSS methods applied to time-delay embedded data as the number of lags are varied using Relief for component selection.

The above experiments show that more lags decrease the performance, but it does not take into account the time difference between lags. When $\tau = 1$, each lagged dimension is only one time sample apart, and depending on the sampling frequency, the values might not be very different. With the sampling frequencies and decimation factors used on each dataset here, the time difference between lagged dimensions is about 8 ms. When $\tau = 1$, two lagged dimensions are only one time sample apart and can be highly correlated. An increase in τ corresponds to lagged dimensions that are more independent. The effects of increasing the time between lags is shown in Figures 4.22, 4.23, and 4.24 as τ is varied from 1 to 20 while using forward selection, ANOVA ranking, and Relief, respectively, for component selection. The number of lags d was set to 6 since it was seen that a large number of lags is not beneficial. In general, the results decrease as τ increases. It is interesting to see that when using ANOVA, the three BSS methods are about the same once τ is greater than 7.



(a) Subject A (b) Subject B (c) Subject C

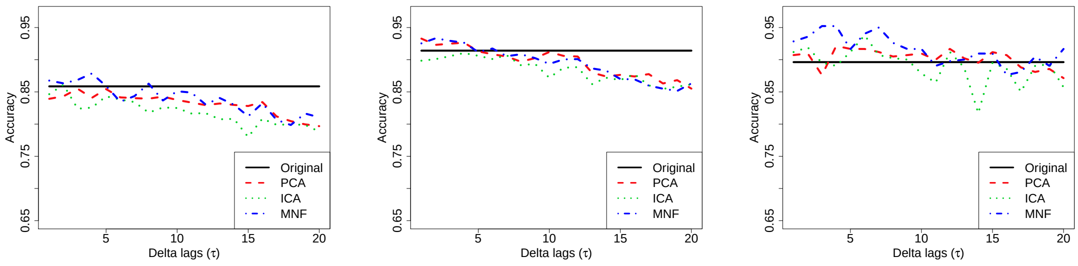
Figure 4.22: Comparison of BSS methods applied to time-delay embedded data as the number of time samples between lagged dimensions are varied using forward selection for component selection.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.23: Comparison of BSS methods applied to time-delay embedded data as the number of time samples between lagged dimensions are varied using ANOVA ranking for component selection.

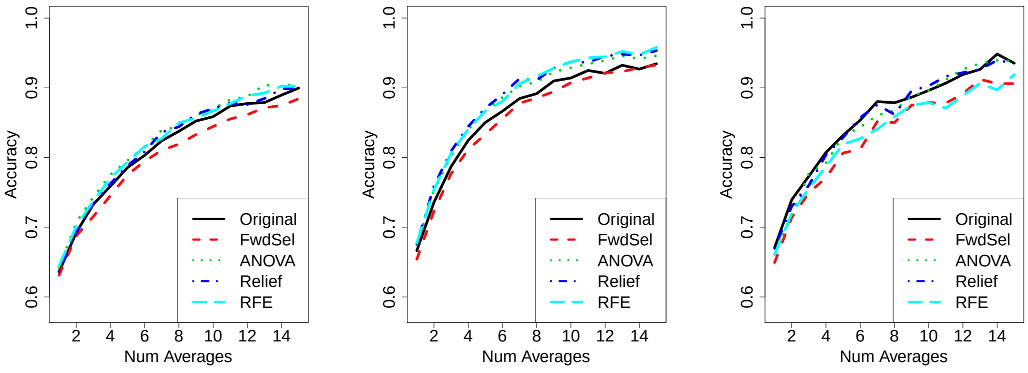
Since RFE was not used in any of the above experiments, Figure 4.25 shows a comparison of all feature selection algorithms after MNF is applied to time-delay embedded data with 15 lags. This



(a) Subject A (b) Subject B (c) Subject C

Figure 4.24: Comparison of BSS methods applied to time-delay embedded data as the number of time samples between lagged dimensions are varied using Relief for component selection.

shows that the performance of RFE is very similar to the other algorithms, suggesting that if RFE was computationally feasible, it would not show any advantages over the other algorithms.



(a) Subject A (b) Subject B (c) Subject C

Figure 4.25: Comparison of feature selection algorithms when using 15 lags with MNF.

Chapter 5

Conclusions

5.1 Summary of Results

The first observation that can be made from these experiments is that it is hard to determine how well these results generalize to other subjects when using only three subjects. Many times, the approaches show different effects depending on the subject. For example, the application of BSS when using all electrodes significantly improves the performance in one subject while decreasing the performance in the other two subjects. Also, there is no feature selection algorithm or BSS method that significantly improves the performance over the baseline for Subject C when using all electrodes, as opposed to the other two subjects. Some performance differences in Subject C can also be attributed to the difference in paradigms used when recording the P300 responses. For example, the paradigm for Subject C evokes a visual response on both target and non-target trials since a letter is flashed for all trials. However, with the P300 speller paradigm used for Subjects A and B, the user focuses their attention on the target letter, and when it flashes, it evokes a visual response in only the target trials. However, with that in mind, there are still many conclusions that can be drawn from these results.

When using all available electrodes, the feature selection algorithms are the main reason for performance gains. At best, the BSS methods (with or without feature selection) achieved similar performance to applying only feature selection, and in most cases, the performance declined. Feature selection algorithms used for channel selection are particularly effective, and this can be explained by the fact that there are many more irrelevant channels when using all electrodes. When considering classification in general, the elimination of irrelevant features will many times cause in increase in performance, which is the case in two of the subjects.

Forward selection and RFE clearly show better performance than either ANOVA ranking or

Relief. This is most likely due to the fact that forward selection and RFE are tied to the SVM algorithm. Whereas Relief and ANOVA are more computationally efficient by estimating the performance of individual features, the feedback from the SVM allows the other two algorithms to achieve better results. Both forward selection and RFE can account for interaction effects between different channels through the use of the SVM. Although RFE is more computationally intensive than forward selection, it only required about twice the amount of time when applied to 64 channels, and therefore, both algorithms are feasible to use.

The effects of feature selection and BSS are reversed on the 8-channel data compared to using all channels. When feature selection is used on the original 8-channel data for channel selection, it does not show any improvements. This is expected because the subset of 8 channels is selected in order to contain the most relevant information. Although feature selection does eliminate some of the channels, it results in similar accuracies to the baseline. The same is true when applied for the task of component selection after BSS. This indicates that, at least for 8-channel data, the BSS methods are not necessarily transforming the data into a set of relevant P300 source components and a set of irrelevant components that should be eliminated. Since the improvements are seen after applying BSS and keeping all source components, the performance gain must be the result of the linear transform of the original data onto a new set of basis vectors. Therefore, the BSS transform increases the amount of relevant information in the source components by applying spatial filters that reduce the mutual information found in adjacent channels.

Across all subjects, the BSS methods perform better on the 8-channel data than on all channels. With fewer channels the P300 source information is more concentrated. However, when using all channels, more information is available that allows the BSS methods to isolate other sources throughout the brain, but the additional information might make the task too complex causing the negative effect seen here. However, it is still surprising that ICA does not benefit from the additional information since it uses higher-order statistics to extract independent sources. In fact, ICA performed considerably worse than both PCA and MNF, which are both based on second-order statistics. One way to look at the relative complexity of these three methods is to look at the time required for training. PCA is the simplest and takes the shortest amount of time, whereas MNF requires an order of magnitude more of training time, and ICA takes another order of magnitude beyond that to extract the source components. One possible explanation for the poor performance of ICA is that it is overfitting the data. Although ICA has proven to be a powerful tool for ERP analysis by Makeig, et al., [28, 29, 30], the application for P300 classification is distinctly different.

When used for ERP analysis, ICA is applied to a very large set of P300 trials where many are averaged together to create a clean, consistent signal in the training data, which allows ICA to find reliable source components. However, when used for P300 classification, it was shown that the best results are obtained when fewer trials are averaged together. In these experiments, five trials were averaged together, which results in a very noisy signal that ICA may be overfitting.

The addition of temporal information in order to allow BSS to extract spatio-temporal components seems to make the problem too complex as well. As more lagged dimensions are included, the performance generally degrades. The problem is especially complicated by the fact that the source components must be projected back to the original data space. Since the final set of selected components are summed together back in the original data space, the individual features associated with each component (used by the feature selection algorithms) are not independent features for the SVM. Instead, the SVM features are comprised of the summation of individual features. This suggests that the application of feature selection algorithms to these individual features might not be the most effective.

The original motivation for this work was to increase the communication rate of the P300 speller application by increasing the classification accuracy when using fewer averaged trials. The best results were obtained from applying BSS on the 8-channel data as shown in Figure 4.12. If a minimum accuracy is desired, of 0.85 for example, then it is possible to look at where the curves intersect this point to find the number of averaged trials required to obtain an accuracy of 0.85. With BSS, Subject A needs only 7 averaged trials to achieve this rather than 10 averaged trials. Both Subjects B and C show a reduction from 6 averaged trials down to 4. This relates to a 43% increase in communication rate in Subject A and a 50% increase in Subjects B and C. This is a significant performance increase, which shows that more advanced software algorithms can help push the P300 speller to become a more practical BCI system.

Overall, these results indicate a set of completely automated EEG preprocessing steps that can best increase the performance, specifically for P300 classification. They show that channel selection is useful when recording from a large number of channels, and BSS is useful when applied to smaller set of electrodes containing information specific to the P300 source signal. This suggests that when recording from a high density electrode system, the first step should be channel selection using forward selection or RFE. It was shown that the best sets of electrodes are different for each subject, which implies that the channel selection step will adapt appropriately to each individual's unique P300 response. When left with only a small number of electrodes, BSS should be applied to obtain a linear transform of the data that results in a set of source components used directly for classification. ICA does not provide any performance advantages over the other methods, and the simplest option of PCA shows the most promise for P300 classification.

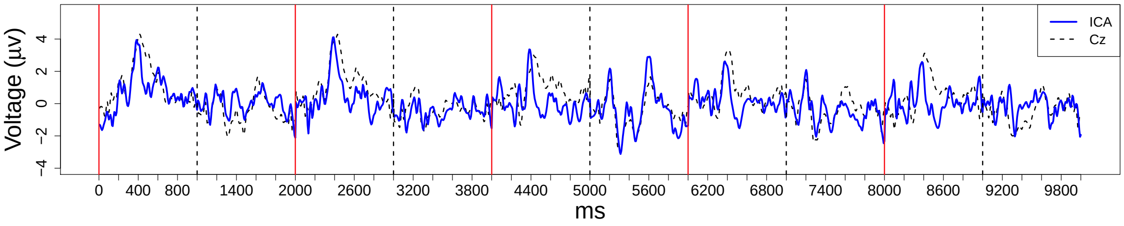
Another interesting result that came from this work but is not directly related to the hypotheses that were explored is that the P300 source signal changed in nature throughout the course of the recording session for Subject C. Figures 5.1 and 5.2 show two different ICA components that are each associated with the positive peak of the P300 signal, depending on whether the trial was from the beginning of the session or the end of the session. 100 trials were averaged together, and the trials were sorted by time in order to capture the overall trend throughout the session. These components show that the spatial distribution of the P300 response changed over time. There are many factors that could account for this change (like fatigue), but the main point is that a user's P300 signal will most likely be different depending on their mental state and their environmental conditions. This presents more evidence for the use of an ensemble of classifiers like Rakotomamonjy and Guigue [16] used that allows the individual classifiers to account for any sort of controlled variability found between trials.

5.2 Future Work

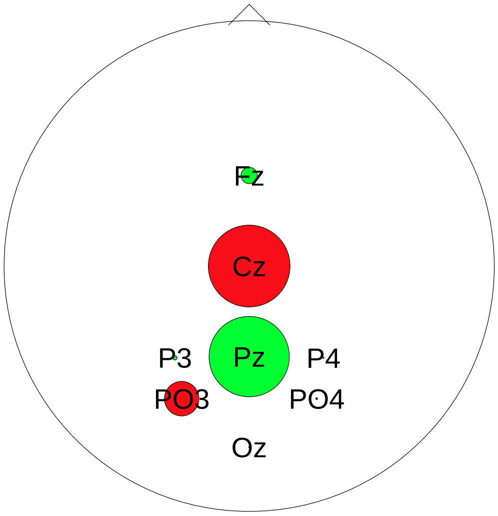
As already mentioned, the same experiments should be run with other subjects to understand how well it generalizes across subjects since the goal is to find a set of preprocessing steps that can applied to any new subject for a performance increase. Along similar lines, this work should be repeated with other classifiers such as linear discriminant analysis (LDA) or neural networks in order to determine if the results generalize well to other classifiers.

There are also other BSS methods that can be explored in this context. Since BSS methods based on second-order statistics showed better results, another approach can be second-order blind identification (SOBI) [51]. Also, common spatial patterns (CSP) [52] is another algorithm based on second-order statistics developed to distinguish between two classes of EEG signals by finding the spatial filters that maximize the variance of one class while minimizing the variance of the other class. It would be interesting to explore the added benefit of using a supervised approach.

Although time-delay embedding ends up complicating the problem, a more thorough analysis could potentially find some benefit. For example, one of the problems seems to be that the components were projected back and summed together. The selection of the most relevant components appears to be the most difficult step, so other approaches for component selection should be explored.

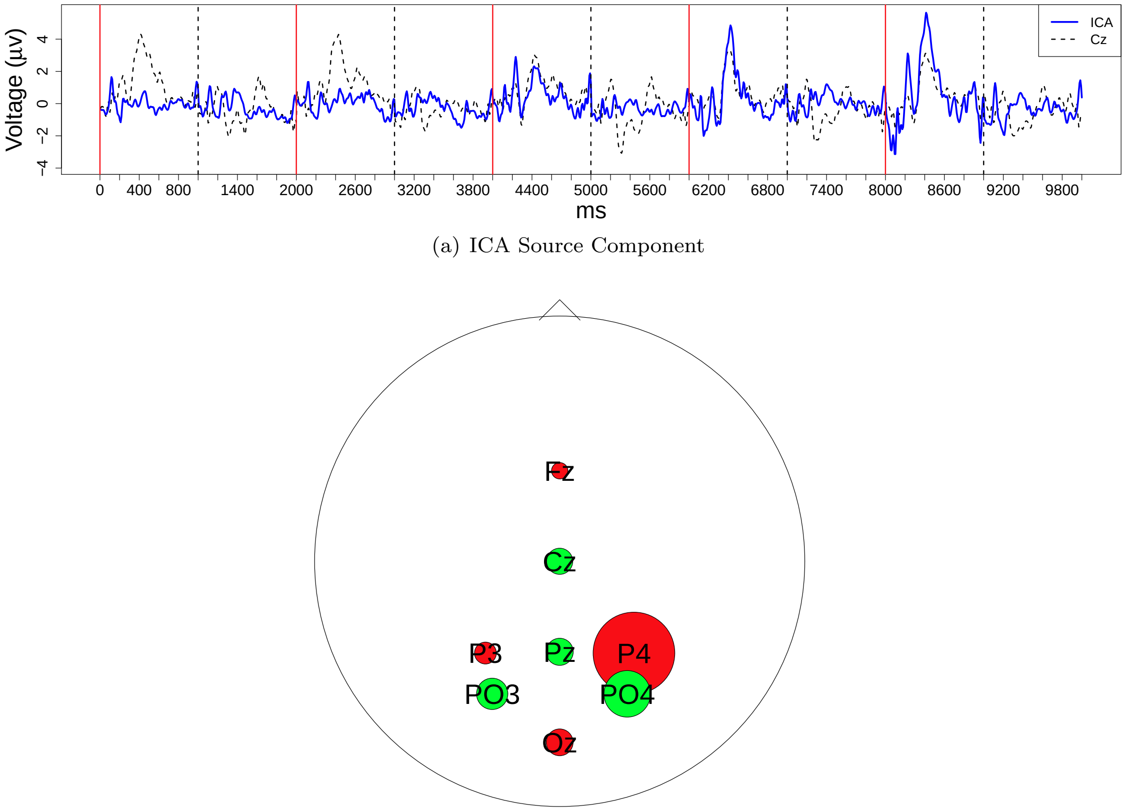


(a) ICA Source Component



(b) Spatial Filter

Figure 5.1: ICA component from Subject C using training data consisting of 100 averaged trials. This component is associated with a larger positive P300 peak from the earlier trials in the session.



(a) ICA Source Component

(b) Spatial Filter

Figure 5.2: ICA component from Subject C using training data consisting of 100 averaged trials. This component is associated with a larger positive P300 peak only from the later trials in the session.

REFERENCES

[1] M. D. Serrua, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, March 2002.

[2] D. M. Taylor, S. I. Helms Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, pp. 1829–1832, 2002.

[3] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol*, vol. 1, p. e42, 10 2003.

[4] J. P. Donoghue, A. Nurmikko, M. Black, and L. R. Hochberg, "Assistive technology and robotic control using motor cortex ensemble-based neural interface systems in humans with tetraplegia," *The Journal of Physiology*, pp. 603–611, March 2007.

[5] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a non-invasive brain-computer interface in humans," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 17849–17854, December 2004.

[6] F. Sharbrough, C. Chatrian, R. Lesser, H. Luders, M. Nuwer, and T. Picton, "American Electroencephalographic Society guidelines for standard electrode position nomenclature," *J. Clin. Neurophysiology*, vol. 8, pp. 200–202, 1991.

[7] A. Furdea, S. Halder, D. J. Krusienski, D. Bross, F. Nijboer, N. Birbaumer, and A. Kbler, "An auditory oddball (P300) spelling system for brain-computer interfaces.," *Psychophysiology*, vol. 46, no. 3, pp. 617–625, 2009.

[8] L. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroenceph. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, 1988.

[9] D. Krusienski, E. Sellers, F. Cabestaing, S. Bayoudh, D. McFarland, T. Vaughan, and J. Wolpaw, "A comparison of classification techniques for the P300 speller," *Journal of Neural Engineering*, vol. 3, no. 4, pp. 299–305, 2006.

[10] B. Blankertz, K.-R. Muller, G. Curio, T. Vaughan, G. Schalk, J. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer, "The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1044–1051, June 2004.

[11] B. Blankertz, K.-R. Müller, D. Krusienski, G. Schalk, J. Wolpaw, A. Schlogl, G. Pfurtscheller, J. Millan, M. Schroder, and N. Birbaumer, "The BCI competition III: validating alternative approaches to actual BCI problems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, pp. 153–159, June 2006.

[12] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter, “BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1073–1076, June 2004.

[13] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, “BCI competition 2003-data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications,” *IEEE Transactions on Biomedical Engineering*, vol. 3, pp. 1067–1072, June 2004.

[14] V. Bostanov, “BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1057–1061, June 2004.

[15] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado, “Ensemble of SVMs for improving brain computer interface P300 speller performances,” in *Artificial Neural Networks: Biological Inspirations - ICANN 2005* (W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, eds.), vol. 3696 of *Lecture Notes in Computer Science*, pp. 45–50, Springer Berlin / Heidelberg, 2005.

[16] A. Rakotomamonjy and V. Guigue, “BCI competition III: Dataset II- ensemble of SVMs for BCI P300 speller,” *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 1147–1154, March 2008.

[17] E. W. Sellers and E. Donchin, “A P300-based brain-computer interface: Initial tests by ALS patients,” *Clinical Neurophysiology*, vol. 117, no. 3, pp. 538–548, 2006.

[18] F. Nijboer, E. Sellers, J. Mellinger, M. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, D. Krusienski, T. Vaughan, J. Wolpaw, N. Birbaumer, and A. Kübler, “A P300-based brain-computer interface for people with amyotrophic lateral sclerosis,” *Clinical Neurophysiology*, vol. 119, no. 8, pp. 1909–1916, 2008.

[19] J. N. Mak, Y. Arbel, J. W. Minett, L. M. McCane, B. Yuksel, D. Ryan, D. Thompson, L. Bianchi, and D. Erdogmus, “Optimizing the P300-based brain-computer interface: current status, limitations and future directions,” *Journal of Neural Engineering*, vol. 8, no. 2, pp. 025003–+, 2011.

[20] K. S. Paulus, I. Magnano, M. R. Piras, M. A. Solinas, G. Solinas, G. F. Sau, and I. Aiello, “Visual and auditory event-related potentials in sporadic amyotrophic lateral sclerosis,” *Clinical Neurophysiology*, vol. 113, no. 6, pp. 853–861, 2002.

[21] T.-P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M. McKeown, V. Iragui, and T. Sejnowski, “Removing electroencephalographic artifacts: comparison between ICA and PCA,” in *Neural Networks for Signal Processing VIII, 1998. Proceedings of the 1998 IEEE Signal Processing Society Workshop*, pp. 63–72, Aug. 1998.

[22] J. Knight, “Signal fraction analysis and artifact removal in EEG,” Master’s thesis, Department of Computer Science, Colorado State University, Fort Collins, CO, 2003.

[23] L. Sun, J. Rieger, and H. Hinrichs, “Maximum noise fraction (MNF) transformation to remove ballistocardiographic artifacts in EEG signals recorded during fMRI scanning,” *NeuroImage*, vol. 46, no. 1, pp. 144–153, 2009.

[24] N. Hill, T. Lal, K. Bierig, N. Birbaumer, and B. Scholkopf, “Attention modulation of auditory event-related potentials in a brain-computer interface,” in *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pp. S3/5/INV – S3/17–20, Dec. 2004.

[25] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, and F. Beverina, “P300-based brain computer interface: Reliability and performance in healthy and paralysed participants,” *Clinical Neurophysiology*, vol. 117, no. 3, pp. 531–537, 2006.

[26] S. Wang and C. J. James, “Enhancing evoked responses for bci through advanced ica techniques,” in *IET 3rd International Conference on Advances in Medical, Signal and Information Processing, 2006. MEDSIP 2006.*, pp. 1–4, July 2006.

[27] K. Li, R. Sankar, Y. Arbel, and E. Donchin, “Single trial independent component analysis for P300 BCI system,” *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pp. 4035–4038, 2009.

[28] S. Makeig, M. Westerfield, T.-P. Jung, J. Covington, J. Townsend, T. J. Sejnowski, and E. Courchesne, “Functionally independent components of the late positive event-related potential during visual spatial attention,” *The Journal of Neuroscience*, vol. 19, no. 7, pp. 2665–2680, 1999.

[29] S. Makeig, M. Westerfield, T.-P. Jung, S. Enghoff, J. Townsend, E. Courchesne, and T. J. Sejnowski, “Dynamic brain sources of visual evoked responses,” *Science*, vol. 295, no. 5555, pp. 690–694, 2002.

[30] S. Makeig and J. Onton, “ERP features and EEG dynamics: An ICA perspective,” in *Oxford Handbook of Event-Related Potential Components*, pp. 1–51, New York, Oxford University Press, March 2009.

[31] T. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf, “Support vector channel selection in BCI,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1003–1010, June 2004.

[32] M. Davies and C. James, “Source separation using single channel ICA,” *Signal Processing*, vol. 87, no. 8, pp. 1819–1832, 2007. Independent Component Analysis and Blind Source Separation.

[33] C. James and S. Wang, “Performance analysis of a P300 BCI speller through single channel ICA,” *Advances in Medical, Signal and Information Processing, 2008. MEDSIP 2008*, pp. 1–4, July 2008.

[34] C. James, D. Abasolo, and D. Gupta, “Space-time ICA versus ensemble ICA for ictal EEG analysis with component differentiation via lempel-ziv complexity,” *Engineering in Medicine and Biology Society, 2007. EMBS 2007*, pp. 5473–5476, Aug. 2007.

[35] M. Davies, C. James, and S. Wang, “Space-time ICA and EM brain signals,” in *Independent Component Analysis and Signal Separation* (M. Davies, C. James, S. Abdallah, and M. Plumbley, eds.), vol. 4666 of *Lecture Notes in Computer Science*, pp. 577–584, Springer Berlin / Heidelberg, 2007.

[36] K. Kira and L. A. Rendell, “The feature selection problem: Traditional methods and a new algorithm,” *Proceedings of AAAI-92*, pp. 129–134, 1992.

[37] D. Krusienski, E. Sellers, D. McFarland, T. Vaughan, and J. Wolpaw, “Toward enhanced P300 speller performance,” *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 15–21, 2008.

[38] S. Makeig, A. Delorme, M. Westerfield, T.-P. Jung, J. Townsend, E. Courchesne, and T. J. Sejnowski, “Electroencephalographic brain dynamics following manually responded visual targets,” *PLoS Biol*, vol. 2, pp. 747–762, June 2004.

[39] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[40] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel, “e1071.” `http://cran.r-project.org/web/packages/e1071/index.html`, 2010.

[41] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[42] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[43] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, "ICA: a potential tool for BCI systems," *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 57–68, 2008.

[44] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, pp. 1483–1492, October 1997.

[45] J. Marchini, C. Heaton, and B. Ripley, "fastICA." `http://cran.r-project.org/web/packages/fastICA/index.html`, 2010.

[46] D. R. Hundley, M. J. Kirby, and M. G. Anderle, "A solution procedure for blind signal separation using the maximum noise fraction approach: Algorithms and examples," in *Proceedings of the Third International Conference on Independent Component Analysis and Signal Separation*, pp. 337–342, 2001.

[47] D. R. Hundley, M. J. Kirby, and M. Anderle, "Blind source separation using the maximum signal fraction approach," *Signal Processing*, vol. 82, no. 10, pp. 1505–1508, 2002.

[48] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, March 2003.

[49] K. J. Johnson and R. E. Synovec, "Pattern recognition of jet fuels: comprehensive GCxGC with ANOVA-based feature selection and principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 60, no. 1-2, pp. 225 – 237, 2002.

[50] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Machine Learning: ECML-94* (F. Bergadano and L. De Raedt, eds.), vol. 784 of *Lecture Notes in Computer Science*, pp. 171–182, Springer Berlin / Heidelberg, 1994.

[51] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, pp. 434–444, Feb. 1997.

[52] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, "Spatial patterns underlying population differences in the background EEG," *Brain Topography*, vol. 2, pp. 275–284, 1990.

Appendix A

Supplemental Figures

Table A.1: Classification accuracy for each possible combination of BSS and feature selection algorithm from Subject A. The mean and standard deviation are shown when using 8 channels and when using all channels.

BSS	Feat sel	Using 8 Channels			Using All Channels		
		1 avg	8 avg	15 avg	1 avg	8 avg	15 avg
Orig	none	.64 ± .011	.84 ± .019	.90 ± .017	.62 ± .014	.80 ± .033	.86 ± .024
ICA	none	.65 ± .012	.87 ± .022	.92 ± .020	.61 ± .008	.78 ± .018	.85 ± .026
MNF	none	.65 ± .014	.88 ± .020	.92 ± .015	.60 ± .009	.77 ± .026	.83 ± .033
PCA	none	.65 ± .013	.87 ± .024	.92 ± .021	.61 ± .010	.78 ± .021	.84 ± .029
Orig	fwdSel	.64 ± .021	.84 ± .037	.90 ± .032	.63 ± .017	.83 ± .029	.88 ± .026
ICA	fwdSel	.65 ± .015	.87 ± .022	.92 ± .023	.61 ± .009	.78 ± .027	.85 ± .036
MNF	fwdSel	.64 ± .021	.85 ± .030	.90 ± .035	.62 ± .013	.80 ± .029	.87 ± .030
PCA	fwdSel	.64 ± .017	.86 ± .030	.91 ± .026	.63 ± .016	.84 ± .028	.90 ± .036
Orig	ANOVA	.63 ± .012	.83 ± .019	.89 ± .019	.60 ± .014	.75 ± .043	.80 ± .047
ICA	ANOVA	.65 ± .015	.87 ± .029	.92 ± .018	.58 ± .014	.72 ± .034	.78 ± .039
MNF	ANOVA	.65 ± .013	.87 ± .026	.91 ± .025	.60 ± .025	.75 ± .058	.81 ± .059
PCA	ANOVA	.64 ± .016	.86 ± .012	.92 ± .023	.61 ± .017	.78 ± .041	.84 ± .041
Orig	Relief	.63 ± .014	.83 ± .023	.88 ± .025	.61 ± .020	.78 ± .045	.84 ± .057
ICA	Relief	.65 ± .014	.87 ± .024	.92 ± .019	.57 ± .011	.69 ± .051	.77 ± .048
MNF	Relief	.65 ± .012	.86 ± .021	.91 ± .015	.60 ± .020	.76 ± .041	.82 ± .039
PCA	Relief	.65 ± .018	.87 ± .024	.92 ± .026	.61 ± .010	.80 ± .018	.85 ± .027
Orig	RFE	.63 ± .016	.84 ± .025	.89 ± .034	.63 ± .007	.85 ± .021	.89 ± .018
ICA	RFE	.65 ± .015	.86 ± .028	.91 ± .024	.61 ± .014	.77 ± .014	.84 ± .023
MNF	RFE	.65 ± .013	.87 ± .018	.92 ± .017	.63 ± .013	.83 ± .032	.88 ± .024
PCA	RFE	.65 ± .012	.86 ± .025	.91 ± .030	.64 ± .016	.84 ± .027	.90 ± .029

Table A.2: Classification accuracy for each possible combination of BSS and feature selection algorithm from Subject B. The mean and standard deviation are shown when using 8 channels and when using all channels.

BSS	Feat sel	Using 8 Channels			Using All Channels		
		1 avg	8 avg	15 avg	1 avg	8 avg	15 avg
Orig	none	.67 ± .010	.89 ± .016	.93 ± .030	.64 ± .009	.84 ± .022	.90 ± .020
ICA	none	.69 ± .011	.93 ± .013	.97 ± .016	.66 ± .011	.89 ± .020	.95 ± .014
MNF	none	.69 ± .012	.93 ± .013	.97 ± .016	.66 ± .010	.89 ± .020	.94 ± .013
PCA	none	.70 ± .008	.93 ± .013	.97 ± .019	.66 ± .012	.88 ± .026	.94 ± .016
Orig	fwdSel	.67 ± .011	.91 ± .019	.95 ± .020	.67 ± .014	.88 ± .017	.93 ± .027
ICA	fwdSel	.69 ± .014	.93 ± .016	.97 ± .013	.65 ± .017	.86 ± .039	.91 ± .043
MNF	fwdSel	.69 ± .009	.93 ± .011	.97 ± .012	.66 ± .014	.89 ± .027	.94 ± .025
PCA	fwdSel	.70 ± .013	.93 ± .013	.96 ± .013	.67 ± .012	.90 ± .031	.94 ± .018
Orig	ANOVA	.67 ± .012	.90 ± .022	.94 ± .021	.62 ± .029	.81 ± .070	.86 ± .078
ICA	ANOVA	.69 ± .011	.93 ± .008	.97 ± .009	.63 ± .029	.82 ± .047	.88 ± .027
MNF	ANOVA	.69 ± .015	.93 ± .013	.97 ± .015	.65 ± .013	.86 ± .032	.92 ± .023
PCA	ANOVA	.69 ± .011	.93 ± .013	.96 ± .014	.64 ± .015	.85 ± .018	.91 ± .031
Orig	Relief	.67 ± .010	.90 ± .022	.94 ± .019	.64 ± .035	.83 ± .051	.89 ± .044
ICA	Relief	.69 ± .012	.93 ± .010	.97 ± .009	.63 ± .015	.83 ± .033	.88 ± .043
MNF	Relief	.69 ± .012	.93 ± .011	.97 ± .018	.64 ± .015	.85 ± .025	.89 ± .030
PCA	Relief	.69 ± .012	.93 ± .011	.96 ± .019	.66 ± .012	.86 ± .032	.92 ± .033
Orig	RFE	.66 ± .013	.90 ± .025	.94 ± .034	.67 ± .013	.90 ± .023	.94 ± .032
ICA	RFE	.69 ± .012	.93 ± .010	.96 ± .014	.66 ± .015	.88 ± .033	.93 ± .025
MNF	RFE	.69 ± .011	.93 ± .013	.97 ± .017	.68 ± .009	.92 ± .020	.96 ± .024
PCA	RFE	.69 ± .009	.93 ± .015	.96 ± .010	.67 ± .010	.90 ± .014	.95 ± .015

Table A.3: Classification accuracy for each possible combination of BSS and feature selection algorithm from Subject C. The mean and standard deviation are shown when using 8 channels and when using all channels.

BSS	Feat sel	Using 8 Channels			Using All Channels		
		1 avg	8 avg	15 avg	1 avg	8 avg	15 avg
Orig	none	.67 ± .025	.88 ± .037	.94 ± .030	.68 ± .025	.89 ± .037	.93 ± .036
ICA	none	.71 ± .031	.93 ± .031	.97 ± .032	.66 ± .025	.87 ± .049	.92 ± .063
MNF	none	.70 ± .030	.92 ± .041	.96 ± .047	.67 ± .026	.87 ± .052	.92 ± .055
PCA	none	.71 ± .025	.93 ± .035	.96 ± .039	.68 ± .022	.86 ± .054	.90 ± .056
Orig	fwdSel	.68 ± .027	.88 ± .038	.94 ± .041	.69 ± .025	.90 ± .048	.96 ± .040
ICA	fwdSel	.69 ± .029	.91 ± .058	.96 ± .040	.65 ± .033	.87 ± .062	.92 ± .056
MNF	fwdSel	.70 ± .026	.91 ± .059	.95 ± .046	.67 ± .033	.86 ± .062	.93 ± .059
PCA	fwdSel	.70 ± .029	.92 ± .055	.94 ± .060	.68 ± .032	.89 ± .052	.92 ± .043
Orig	ANOVA	.68 ± .029	.88 ± .032	.94 ± .047	.66 ± .021	.86 ± .037	.93 ± .057
ICA	ANOVA	.69 ± .038	.93 ± .029	.97 ± .032	.67 ± .035	.86 ± .061	.91 ± .072
MNF	ANOVA	.71 ± .028	.92 ± .036	.97 ± .033	.65 ± .039	.82 ± .049	.89 ± .061
PCA	ANOVA	.70 ± .031	.91 ± .033	.96 ± .030	.68 ± .025	.88 ± .048	.93 ± .065
Orig	Relief	.68 ± .025	.90 ± .053	.94 ± .047	.68 ± .028	.89 ± .033	.95 ± .035
ICA	Relief	.71 ± .029	.92 ± .024	.97 ± .031	.64 ± .038	.81 ± .064	.86 ± .069
MNF	Relief	.70 ± .026	.92 ± .040	.96 ± .049	.64 ± .036	.80 ± .062	.88 ± .078
PCA	Relief	.70 ± .025	.92 ± .046	.95 ± .031	.71 ± .025	.92 ± .034	.96 ± .040
Orig	RFE	.68 ± .026	.90 ± .037	.94 ± .046	.68 ± .021	.91 ± .037	.95 ± .035
ICA	RFE	.70 ± .026	.92 ± .043	.95 ± .035	.68 ± .027	.87 ± .042	.92 ± .042
MNF	RFE	.70 ± .030	.92 ± .033	.95 ± .043	.66 ± .029	.85 ± .044	.91 ± .058
PCA	RFE	.71 ± .039	.93 ± .050	.98 ± .035	.71 ± .028	.90 ± .046	.94 ± .041